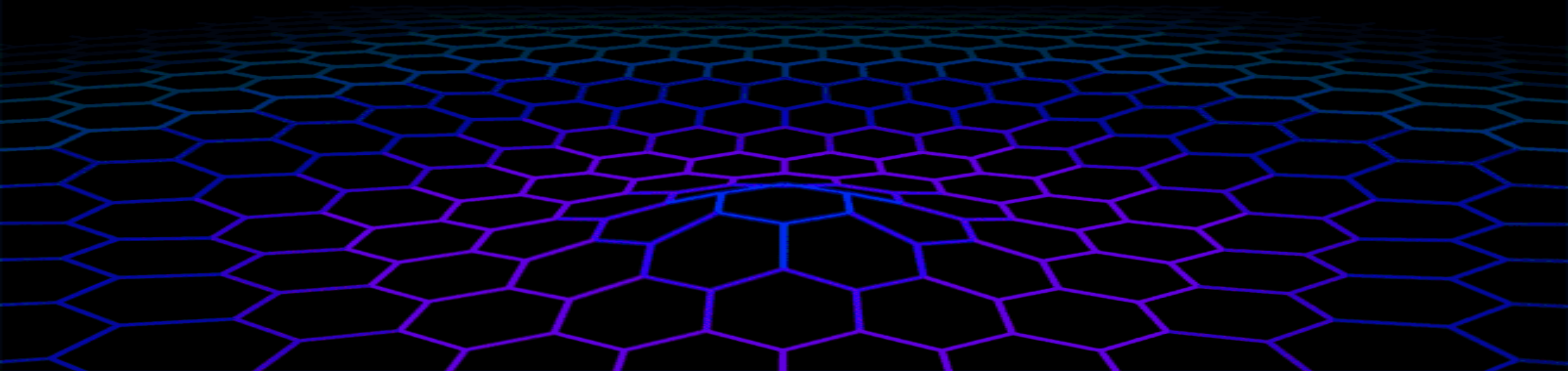


# Fundamentos Computacionais



E agora,  
o que a  
gente faz?



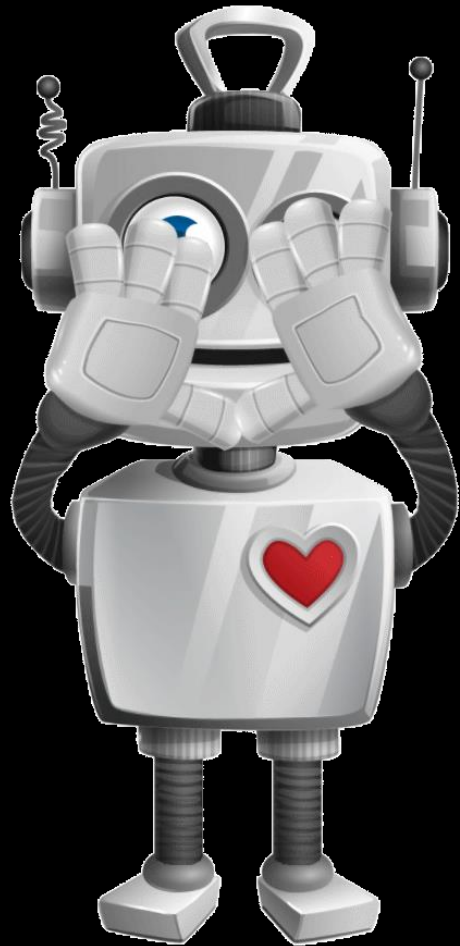
Faz.



TÉO &  
MINI MUNDO

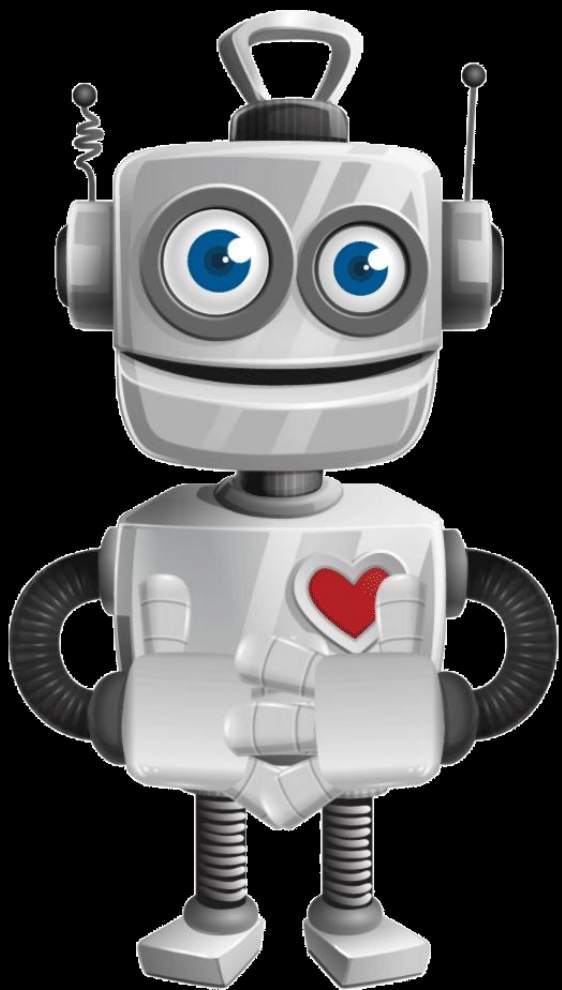
CAETANO CURY

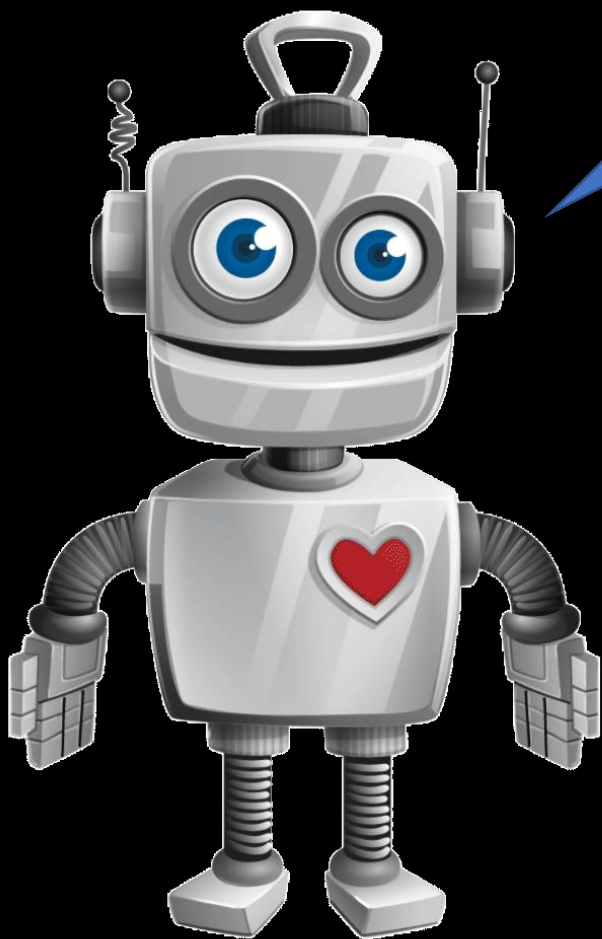
Oi!  
Você sabia que quando  
entramos na internet,  
podemos ver várias páginas  
diferentes?





Sério?  
Eu não sabia disso.  
Como essas páginas são feitas?

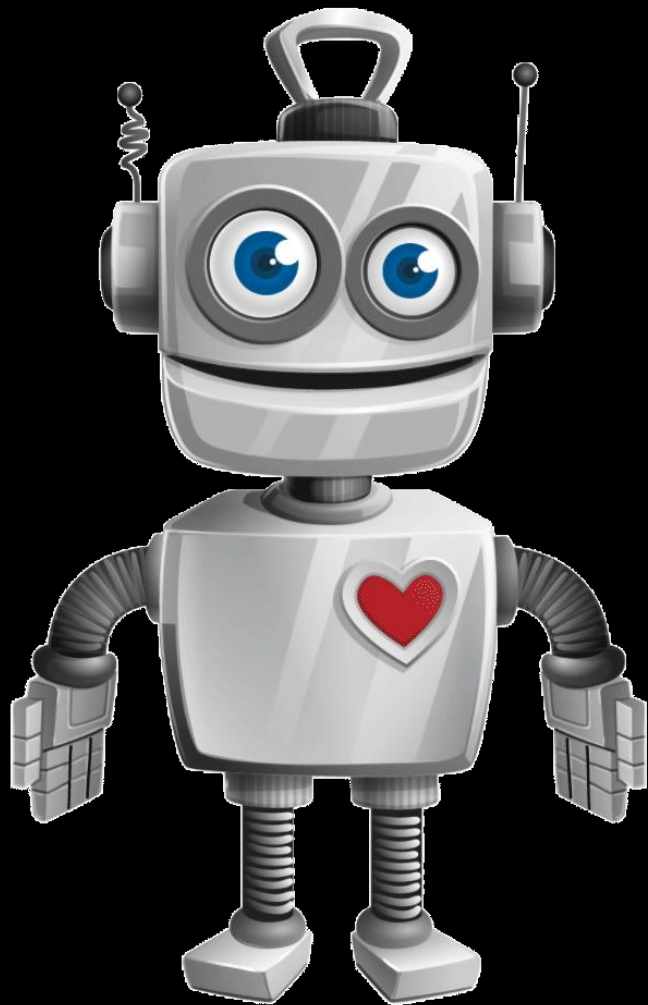




Elas são feitas usando  
linguagens de programação.  
Uma delas é o **JavaScript**.

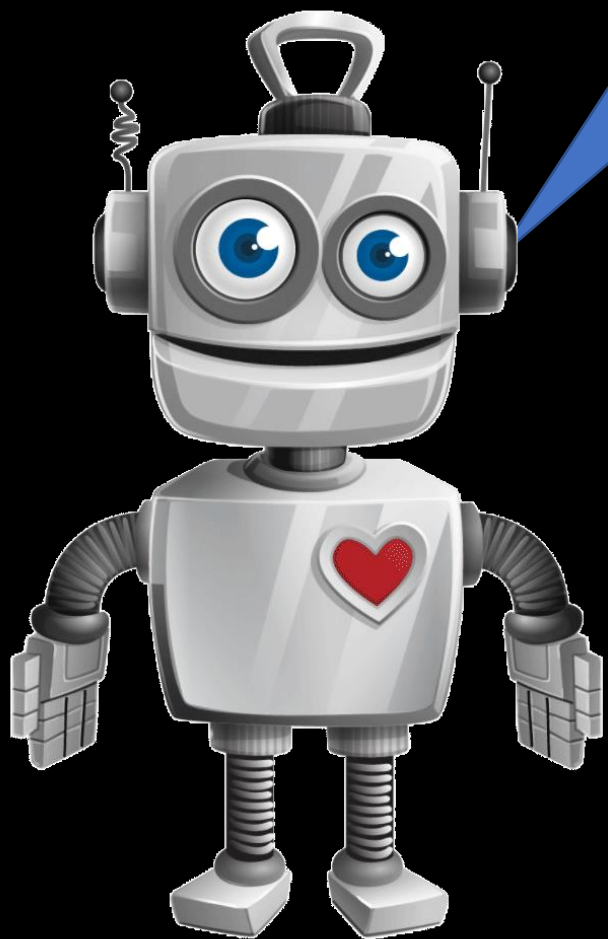
Eu aprendi sobre isso com o **Tio  
Dedé**, também conhecido como  
**Dr. Edecio**.





Ah, o **JavaScript**!  
Eu já ouvi falar.  
É uma linguagem de programação  
bem poderosa, né?





Exatamente!  
O JavaScript é **bem legal** (me sinto  
o Edecio falando assim, hahaha).

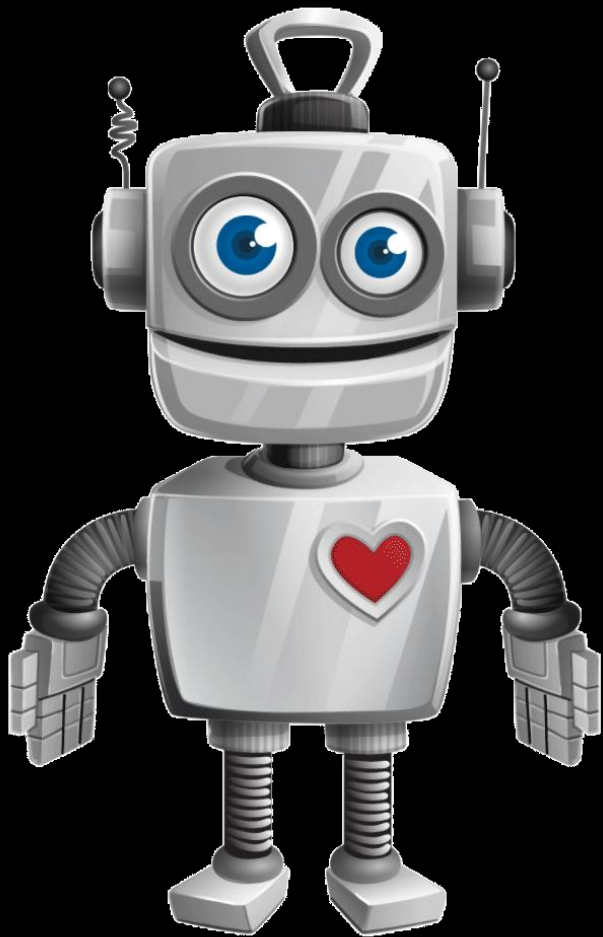
Com ele, nós, os programadores  
maravilhosos, podemos criar um  
monte de coisas legais na web.



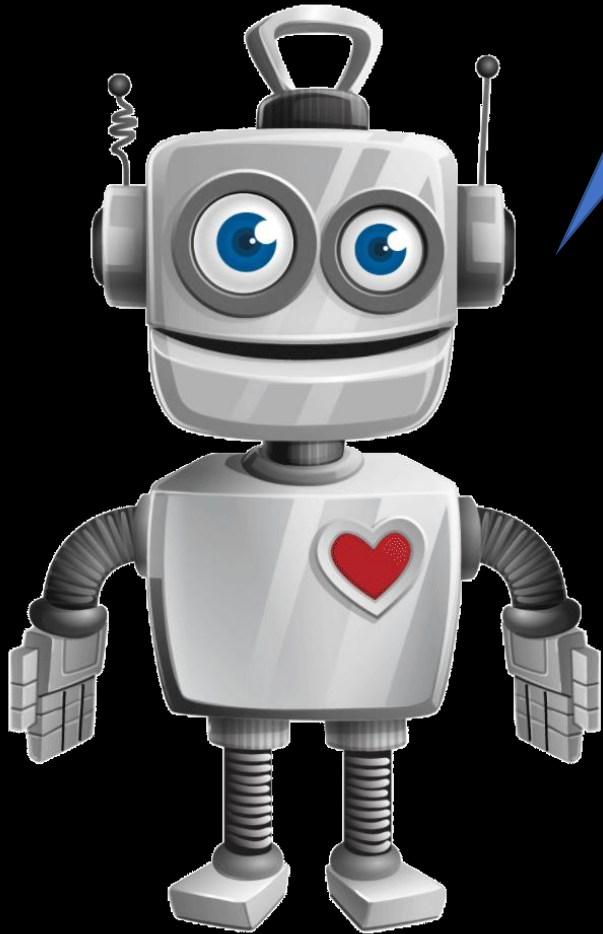
Nossa, deve ser incrível poder  
fazer essas coisas.

Mas e o Node.js?

O que é isso?







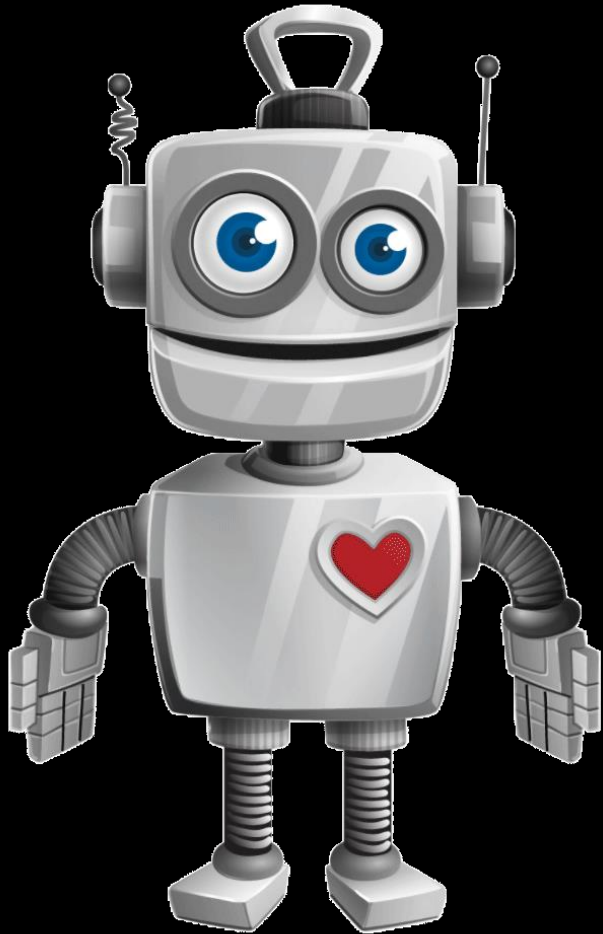
Ah, o Node.js é uma coisa  
superlegal.

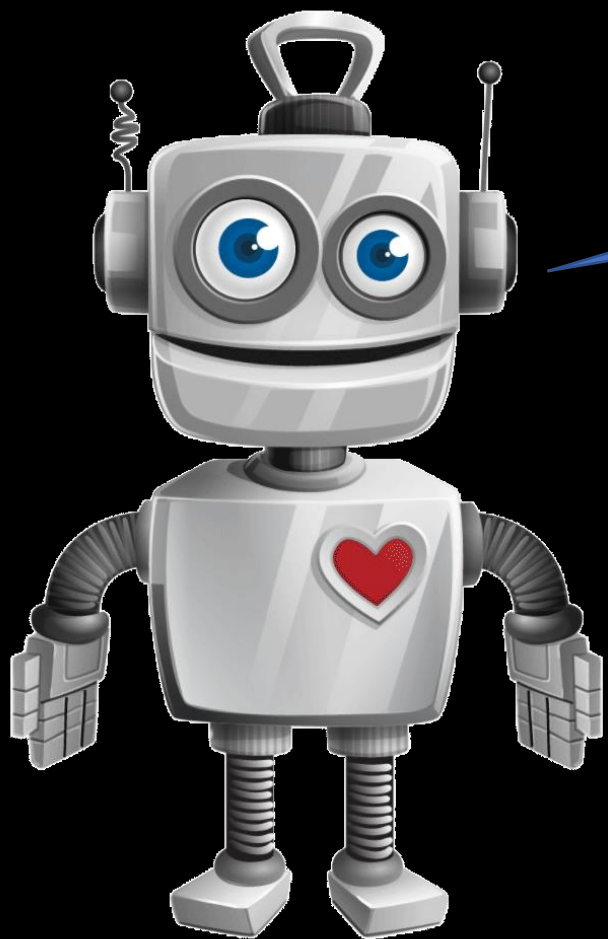
Ele foi criado para fazer o  
JavaScript funcionar fora dos  
navegadores, nos servidores.



Servidores?

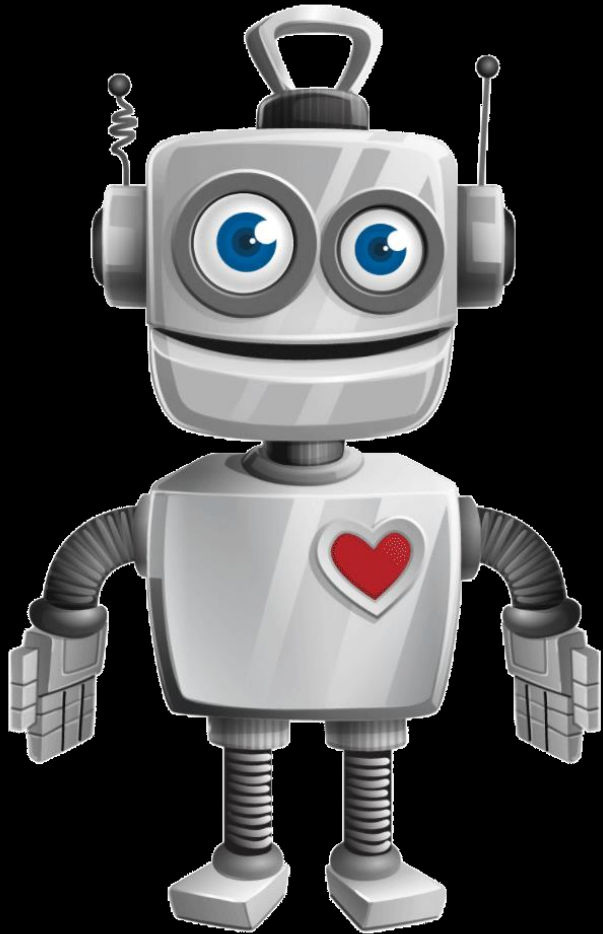
O que são servidores?





Servidores são como computadores poderosos que armazenam e enviam as páginas da web para as pessoas quando elas pedem.  
Sabe o nosso amigo **Mussum**?  
Ele é um servidor!

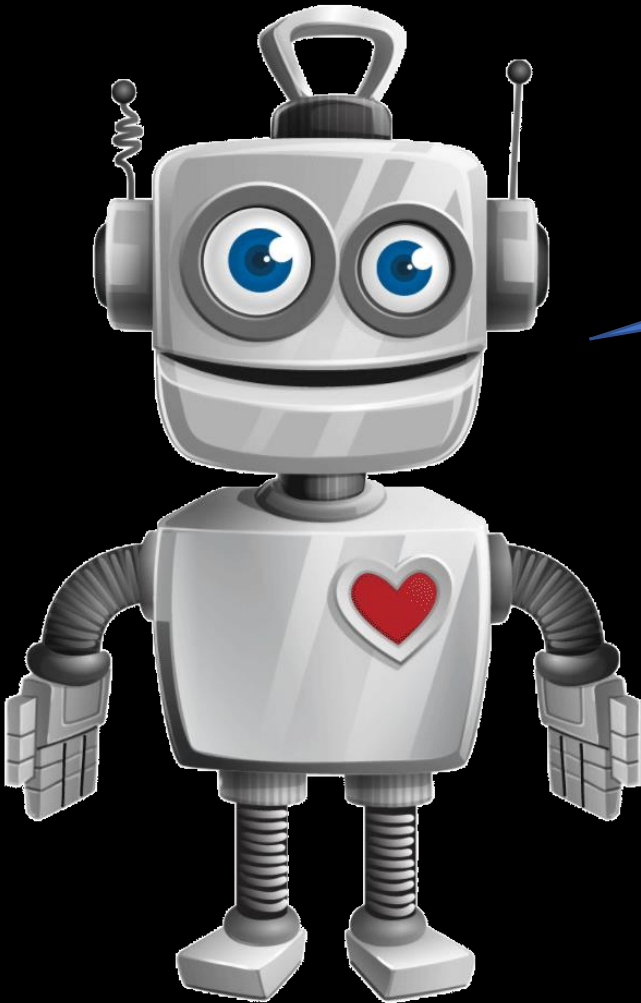




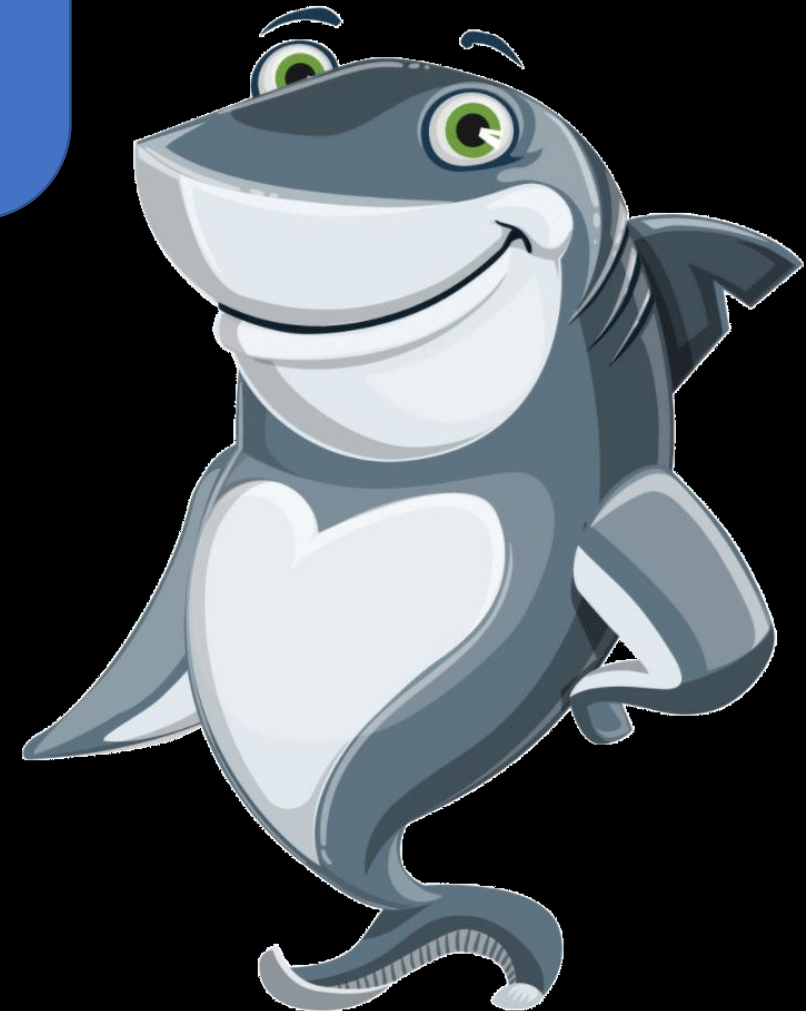
Uau, que legal!  
Então, o Node.js permite que os  
programadores usem o JavaScript  
para criar servidores e aplicativos  
da web, né?

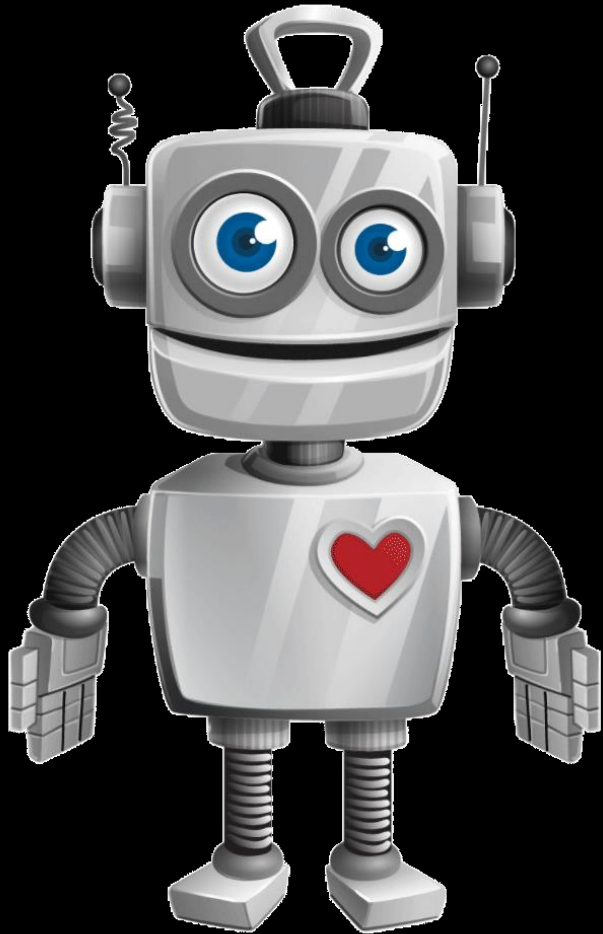






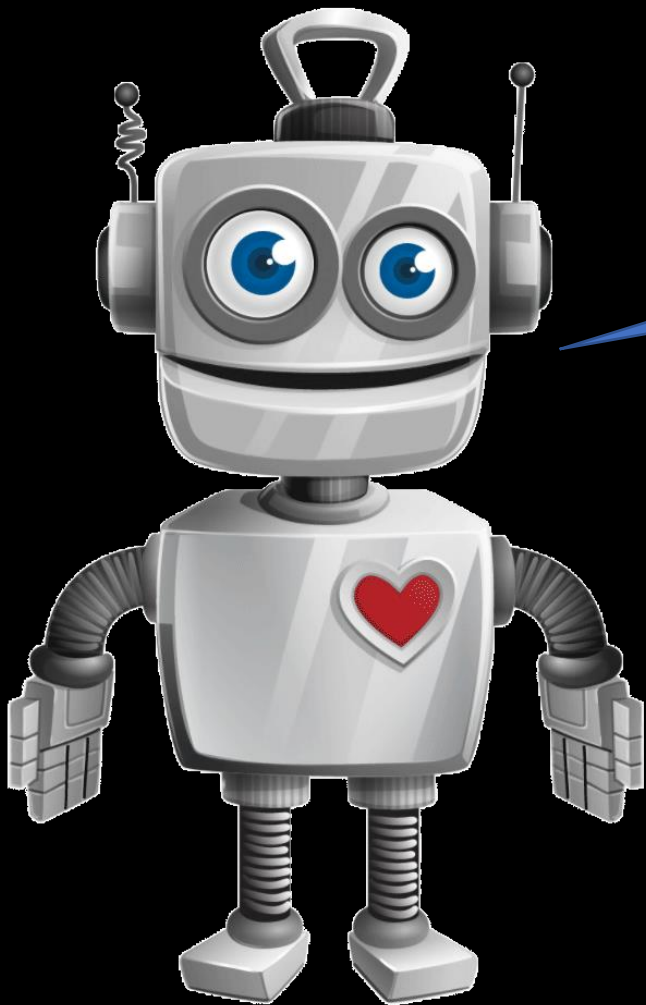
Exatamente!  
Com o Node.js, a gente consegue  
escrever o código dos servidores e  
criar coisas bem bacanas na  
internet usando o JavaScript.





Isso é tri demais!  
Agora eu entendi melhor o  
que é o Node.js.  
É uma ferramenta incrível  
para os programadores  
fazerem coisas legais na web.

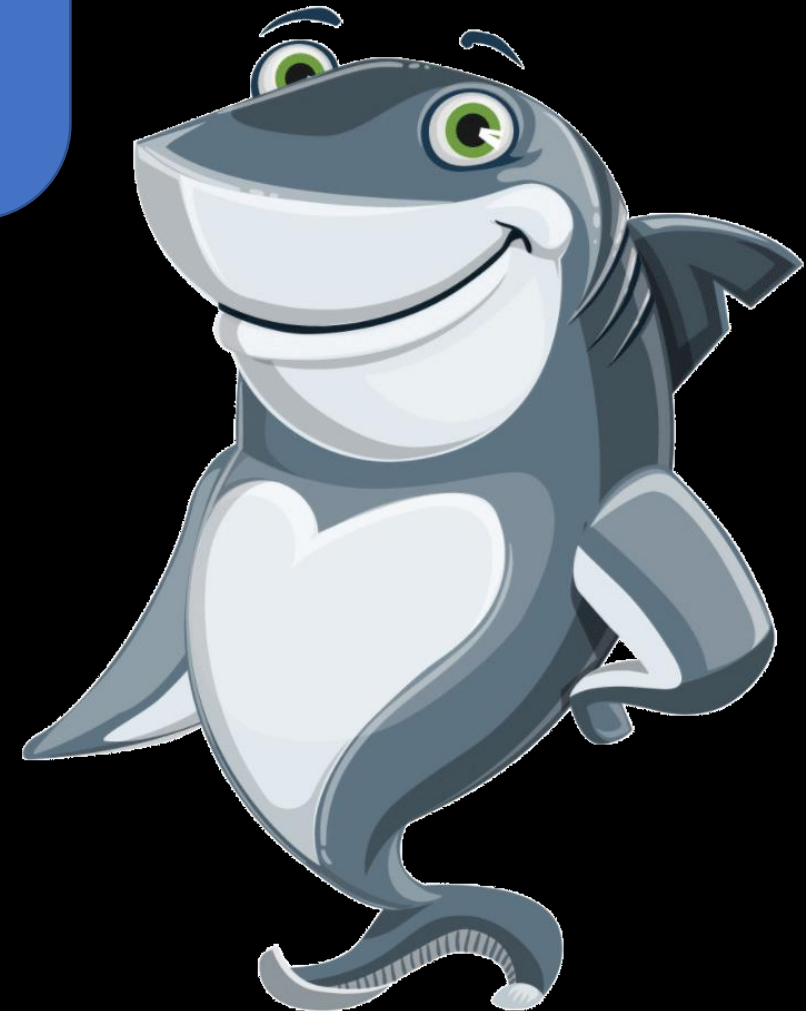




Com certeza!

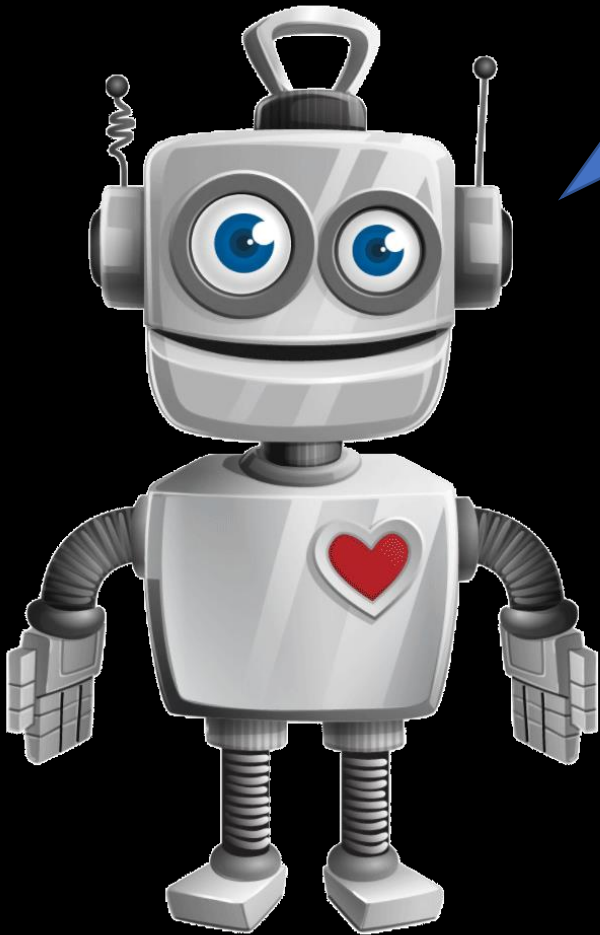
O Node.js é realmente bacaninha.

É uma forma de usar o JavaScript  
para fazer coisas incríveis na  
internet.

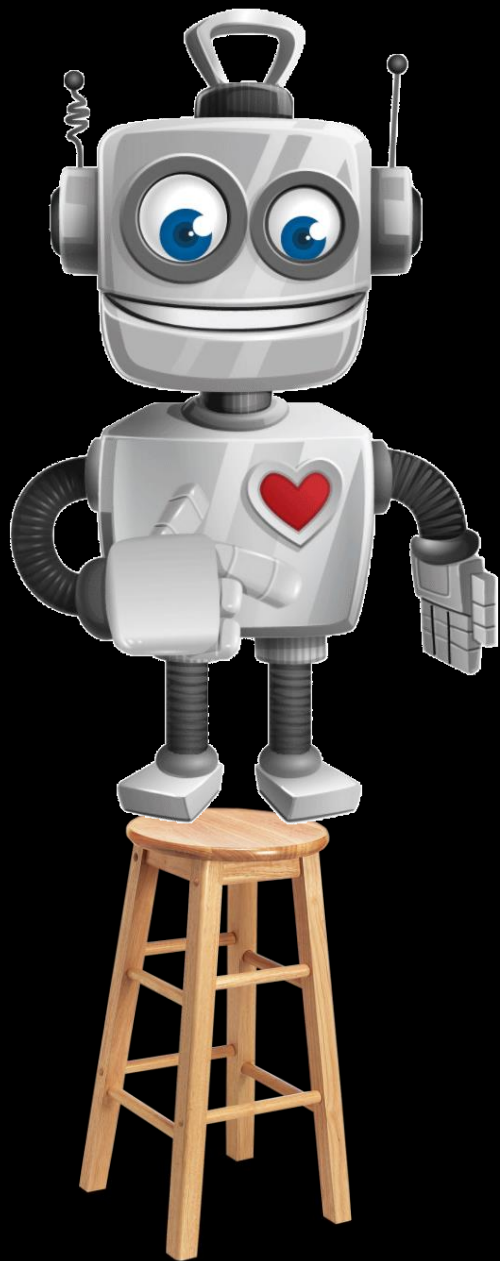


Que tal usar alguns conceitos de Fundamentos Computacionais (que a gente viu com o **Tio Glad**) na programação?

Com **JavaScript** e **Node.js**, claro.







Ah!  
**Conjunção, Disjunção,  
Negação e Vera Fisher?**  
Sou fissurado naquela coroa.  
Bora!

# Criando uma pasta para trabalhar

C:

CD \

MD fc

CD fc

```
PowerShell 7 Preview
PS C:\> c:
PS C:\> cd \
PS C:\> md fc

Directory: C:\

Mode                LastWriteTime         Length Name
----                -
d-----         19/06/2022   13:31             fc

PS C:\> cd fc
PS C:\fc> |
```



# Node.js

Interpretador Javascript

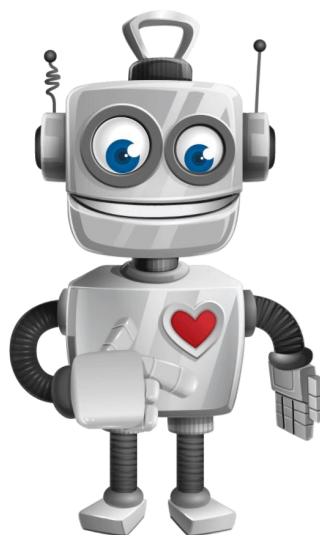
Não depende do navegador

Usa o V8 “motor” interpretador de Javascript (criado por Google e Libuv)

Torna o Javascript uma linguagem capaz de criar programas desktop.

## Instalação:

Acesse <https://nodejs.org/en/download/> (verifique a versão compatível com seu S.O.)



# prompt-sync

Módulo que permite aceitar a entrada do usuário do console NodeJS.

**Obs.:** Este método é síncrono (a instância do Node espera a entrada antes de executar a linha seguinte)

**Instalação:**

**npm** install prompt-sync

ou

**yarn** add prompt-sync

```
PS C:\fc> npm install prompt-sync
added 3 packages, and audited 4 packages in 800ms
found 0 vulnerabilities
PS C:\fc>
```



# prompt-sync

Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda

EXPLORADOR

- FC
  - node\_modules
    - ansi-regex
    - prompt-sync
    - strip-ansi
    - .package-lock.json
    - package-lock.json
    - package.json
    - triangulo.js
    - vaga.js

PROBLEMAS SAÍDA TERMINAL SQL CONSOLE CONSOLE DE DEPURAÇÃO

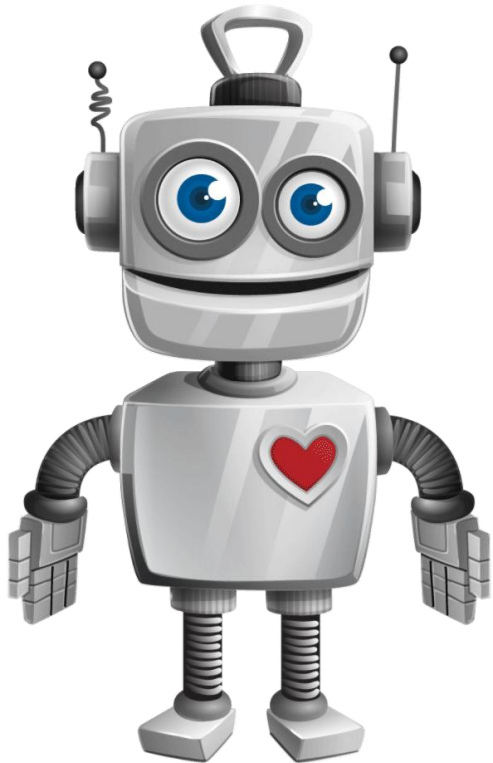
```
PS C:\fc> npm i prompt-sync

up to date, audited 4 packages in 1s

found 0 vulnerabilities
PS C:\fc>
```



# Programas que utilizam conectivos lógicos



&&	$\wedge$	Conjunção	AND	E
	$\vee$	Disjunção	OR	OU
!	$\sim$	Negação	NOT	NÃO

# Exemplo1

Elaborar um programa que leia **nome**, **idade** e **pretensão salarial** de um candidato a uma vaga de emprego.

Caso a idade esteja entre 20 e 30 anos e a pretensão salarial entre 2000.00 e 3000.00 exiba “<nome> você foi selecionado” ou senão, “<nome> procure outra vaga”.



# Exemplo1

Elaborar um programa que leia **nome**, **idade** e **pretensão salarial** de um candidato a uma vaga de emprego.

Caso a idade esteja entre 20 e 30 anos e a pretensão salarial entre 2000.00 e 3000.00 exiba “<nome> você foi selecionado” ou senão, “<nome> procure outra vaga”.

## Observação

- Utilize o **if** para exibir a mensagem de “<nome> você foi selecionado”
- Utilize o **else** para exibir a mensagem de “<nome> procure outra vaga”



# Exemplo1

Elaborar um programa que leia **nome**, **idade** e **pretensão salarial** de um candidato a uma vaga de emprego.

Caso a idade esteja entre 20 e 30 anos e a pretensão salarial entre 2000.00 e 3000.00 exiba “<nome> você foi selecionado” ou senão, “<nome> procure outra vaga”.

## Observação

- Utilize o **if** para exibir a mensagem de “<nome> você foi selecionado”
- Utilize o **else** para exibir a mensagem de “<nome> procure outra vaga”
- **Desafio:** Após terminar, “comente” as linhas **if** e **else** e defina um novo **if** e **else**. Agora, o novo **if** deve conter a condição para exibir a mensagem “<nome> procure outra vaga” e o **else** “<nome> você foi selecionado”.

# Exemplo1

Elaborar um programa que leia **nome**, **idade** e **pretensão salarial** de um candidato a uma vaga de emprego.

Caso a idade esteja entre 20 e 30 anos e a pretensão salarial entre 2000.00 e 3000.00 exiba “<nome> você foi selecionado” ou senão, “<nome> procure outra vaga”.

## Observação

- Utilize o **if** para exibir a mensagem de “<nome> você foi selecionado”
- Utilize o **else** para exibir a mensagem de “<nome> procure outra vaga”
- **Desafio:** Após terminar, “comente” as linhas **if** e **else** e defina um novo **if** e **else**. Agora, o novo **if** deve conter a condição para exibir a mensagem “<nome> procure outra vaga” e o **else** “<nome> você foi selecionado”.

# Enunciado revisado (Gladimir)

Elaborar um programa em JavaScript que leia o nome, idade e pretensão salarial de um candidato a uma vaga de emprego. O programa deve seguir as seguintes etapas:

Solicitar ao usuário que digite seu nome, idade e pretensão salarial.

Verificar se a idade está entre 20 e 30 anos e se a pretensão salarial está entre 2000.00 e 3000.00.

Se a idade estiver no intervalo e a pretensão salarial estiver dentro do limite, exibir a mensagem "<nome>, você foi selecionado".

Caso contrário, exibir a mensagem "<nome>, procure outra vaga".

Após a conclusão dessas etapas, você deve realizar o desafio proposto:

Comentar as linhas do if e do else anteriores.

Definir um novo if e um novo else.

No novo if, a condição deve ser para exibir a mensagem "<nome>, procure outra vaga".

No novo else, a mensagem deve ser "<nome>, você foi selecionado".

# Exemplo1

Ler nome, idade e pretensão salarial

**if** ... idade entre 20 e 30      pretensão salarial entre 2000.00 e 3000.00

“<nome> você foi selecionado”

**else** ...

“<nome> procure outra vaga”.

**Desafio:**

**if** ?

“<nome> procure outra vaga”

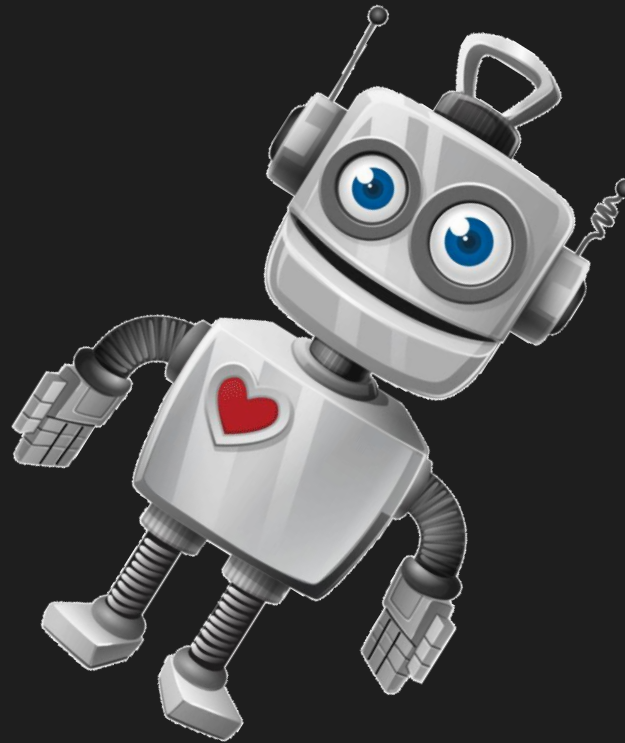
**Else**

“<nome> você foi selecionado”.





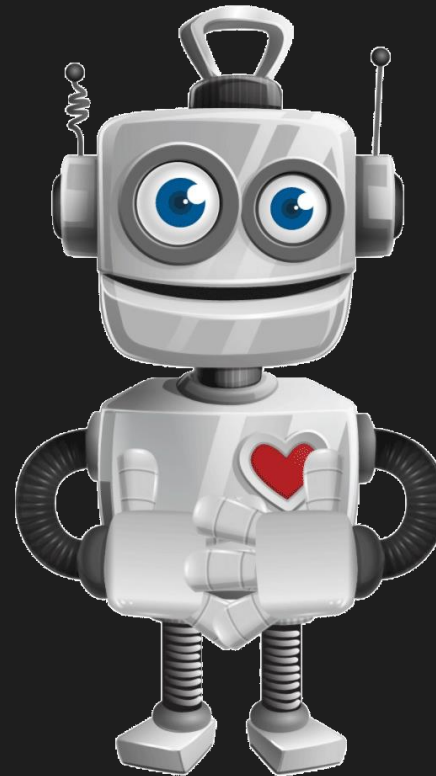
```
JS vaga.js > ...  
1  /* Elaborar um programa que leia nome, idade e pretensão salarial de um candidato a uma vaga de emprego.  
2  Caso a idade esteja entre 20 e 30 anos e a pretensão salarial entre 2000.00 e 3000.00  
3  exiba "<nome> você foi selecionado" ou senão, "<nome> procure outra vaga". */
```



Enunciado  
“comentado” para que  
não seja interpretado  
durante a execução

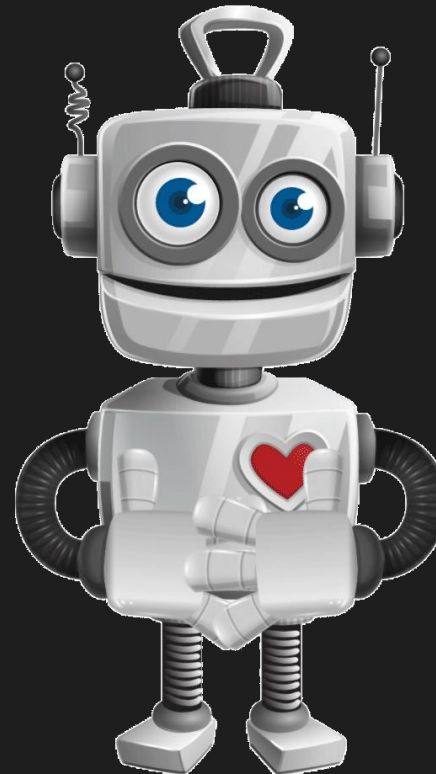
```
JS vaga.js > ...  
1  /* Elaborar um programa que leia nome, idade e pretensão salarial de um candidato a uma vaga de emprego.  
2  Caso a idade esteja entre 20 e 30 anos e a pretensão salarial entre 2000.00 e 3000.00  
3  exiba "<nome> você foi selecionado" ou senão, "<nome> procure outra vaga". */  
4  
5  const prompt = require("prompt-sync")()
```

Vamos usar o  
prompt-sync



```
JS vaga.js > ...
1  /* Elaborar um programa que leia nome, idade e pretensão salarial de um candidato a uma vaga de emprego.
2  Caso a idade esteja entre 20 e 30 anos e a pretensão salarial entre 2000.00 e 3000.00
3  exiba "<nome> você foi selecionado" ou senão, "<nome> procure outra vaga". */
4
5  const prompt = require("prompt-sync")()
6
7  let nome = (prompt("Nome: "))
8  let idade = Number(prompt("Idade: "))
9  let pret = Number(prompt("Pretensão salarial: "))
```

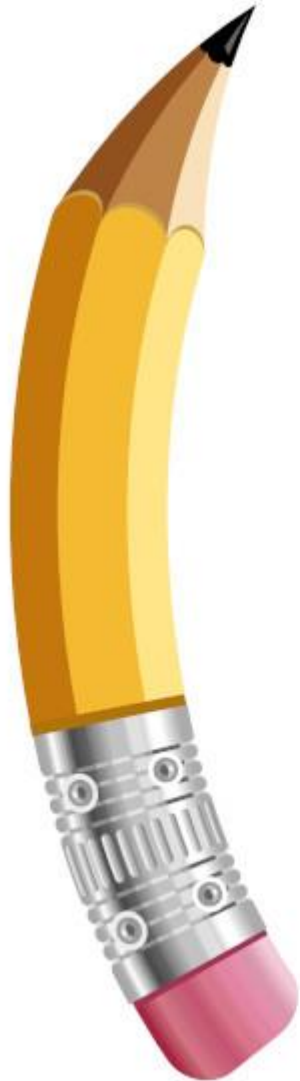
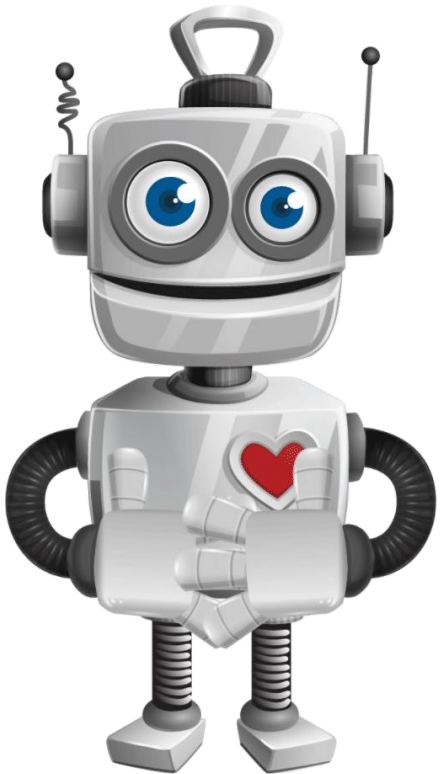
Declarando  
variáveis com **let**



# Abrindo parenteses para falar sobre VAR, LET e CONST

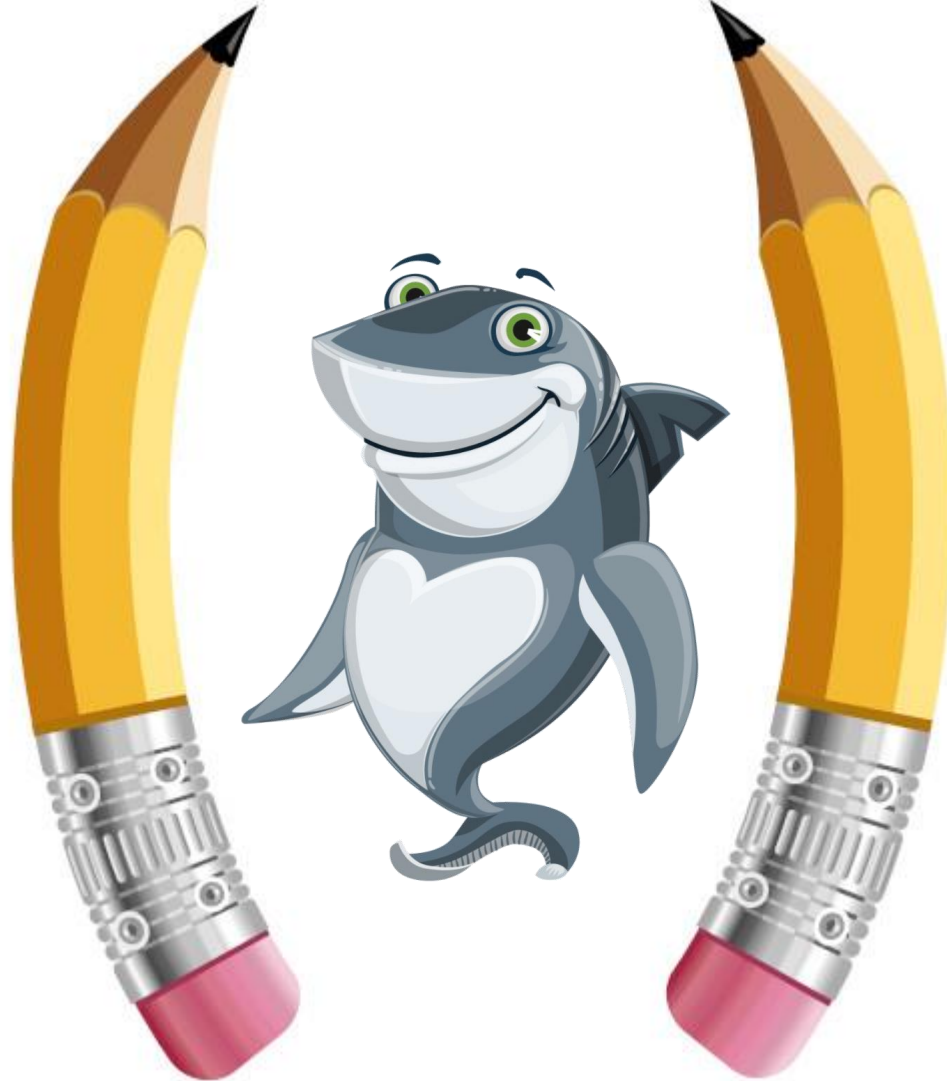
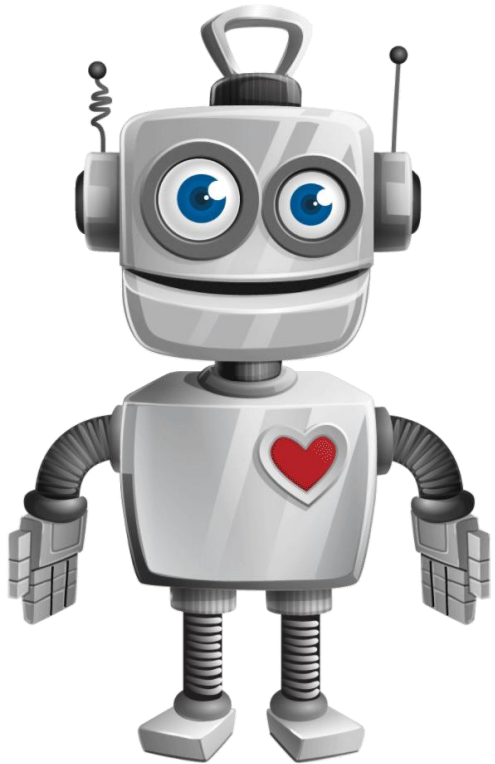


# Abrindo parenteses para falar sobre VAR, LET e CONST

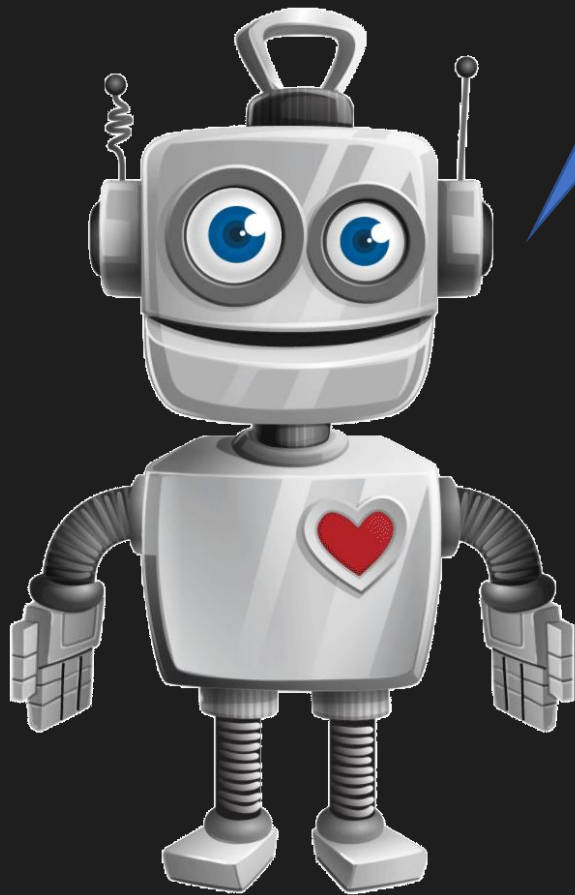




# Abrindo parenteses para falar sobre VAR, LET e CONST



Declarando variáveis com  
VAR, LET e CONST



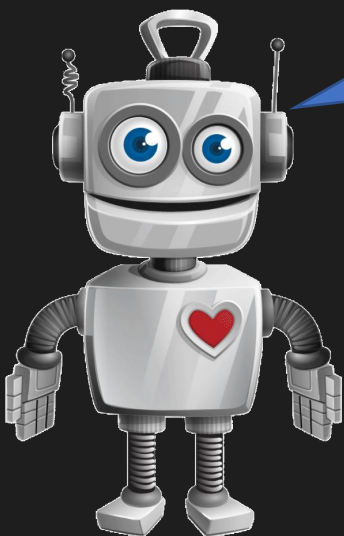
# VAR, LET e CONST

Enquanto eu recarrego as minhas baterias na cantina, a Turma da Mônica vai demonstrar alguns exemplos nos próximos slides.

Preste bastante atenção pois eles não vão repetir (hahahahah)

Você prefere hahahaha ou kkkkkk?

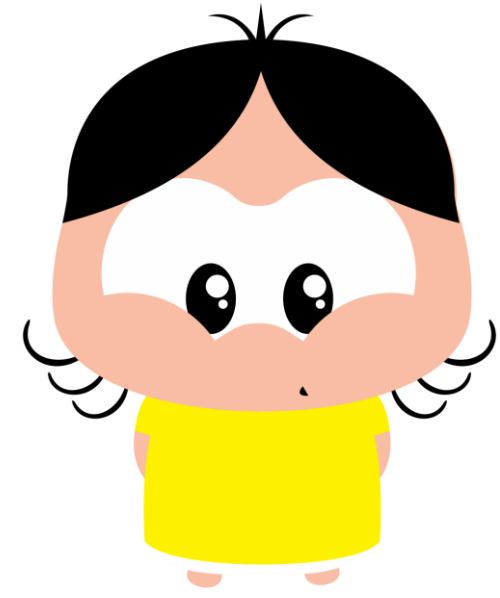
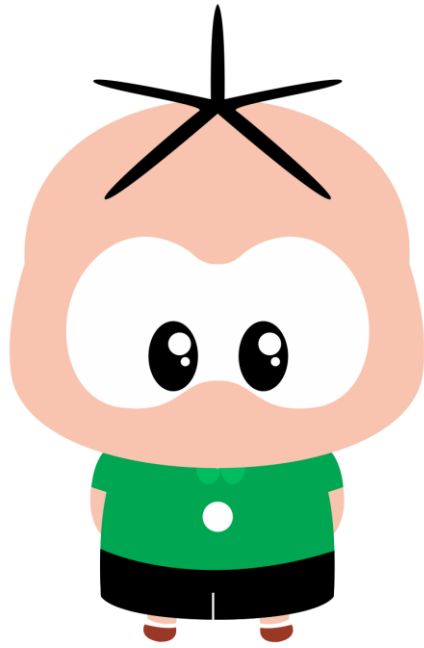
Fui!



# VAR, LET e CONST

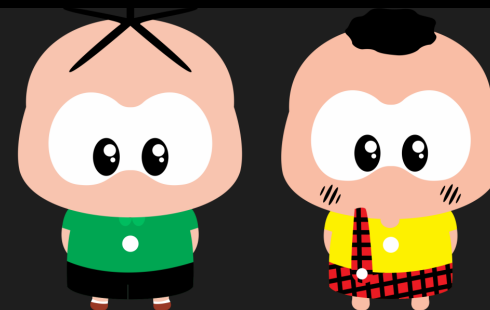


# Presta atenção!





# Veja o escopo do VAR



```
var cebolinha = "Vamos blincar";
```

```
function funcao() {  
    var cascao = "Odeio chuva";  
    console.log(cascao); // Odeio chuva  
}
```

```
funcao();
```

```
console.log(cebolinha); // Vamos blincar  
console.log(cascao);    // ReferenceError: cascao is not defined
```

# Posso redeclarar e redefinir VAR

```
var monica = "Dou coelhada mesmo";  
var monica = "Sansão me ajuda";  
var monica = "Tô nem aí";  
monica = "Quer levar coelhada também?";
```



# Hoisting

```
// Se escrever assim  
console.log (monica); // undefined  
var monica = "Cadê o Sansão?"
```

O “motor” JavaScript move todas as declarações de variáveis e de funções para o topo do seu código. Esse recurso é conhecido como hoisting (içamento / elevação) JavaScript.



```
// O interpretador vai entender assim  
var monica;
```

```
console.log(monica); // undefined  
monica = "Cadê o Sansão?";
```

# Ponto de atenção com o VAR



```
var monica = "Não quero brincar";  
var contador = 4;  
  
if (contador > 3) {  
    var monica = "Vamos brincar";  
    console.log(monica); // Vamos brincar  
}  
console.log(monica)      // "Vamos brincar"
```

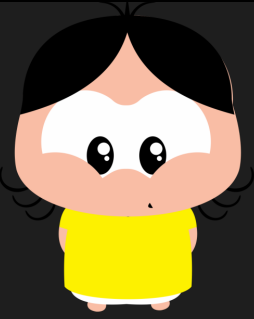
# LET tem escopo de bloco



```
let contador = 3;
if (contador > 2) {
  let cascao = "Não posso me molhar";
  console.log(cascao); // "Não posso me molhar"
}
console.log(cascao) // ReferenceError: cascao is not defined
```



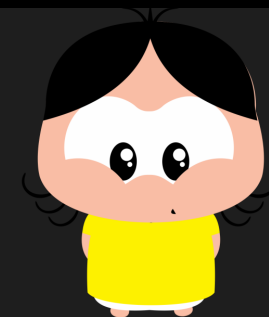
# LET sendo redefinido em escopos diferentes



```
let magali = "Quero um cheese Bebum com ovo";

if (true) {
  let magali = "Quero um Redsburger com bacon";
  console.log(magali); // "Quero um Redsburger com bacon"
}
console.log(magali);    // "Quero um cheese Bebum com ovo"
```

# LET



```
// LET pode ser redefinido  
let magali = "Estou com fome";  
magali = "Estou com fome e sede";
```

```
// LET não pode ser redeclarado  
let magali = "Quero melancia";  
let magali = "Meu gato se chama Mingau"; // SyntaxError: Identifier  
'magali' has already been declared
```

# CONST

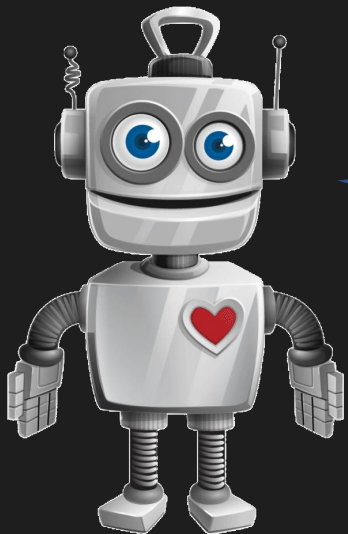


```
// CONST não pode ser redeclarado
const astronauta = "Eu vivo sempre no mundo da lua";
const astronauta = "Foguete dá ré sim"; // SyntaxError: Identifier 'astronauta'
has already been declared
```

```
// CONST não pode ser redefinido
const astronauta = "Tem muito espaço no espaço";
astronauta = "O som não propaga no vácuo"; // TypeError: Assignment to constant
variable.
```

# Declarando variáveis com LET

```
console.log(magali); // ReferenceError: Cannot access 'magali' before initialization  
let magali = "Chocolate é vida";
```



Voltei!

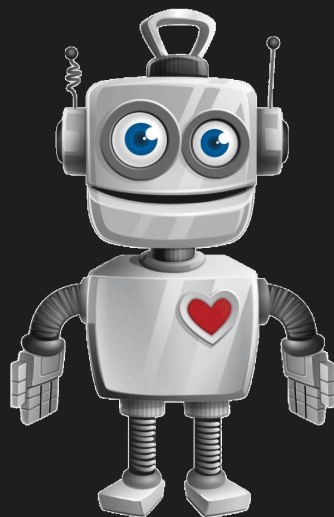
A mensagem desse erro diz que a variável contador já está na memória heap, mas não foi inicializada.

O motor JavaScript “levanta” as declarações de variáveis que usam a palavra-chave **LET**. Mas ele não inicializa essas variáveis.

# Tabelinha VAR, LET e CONST

Obs.: VAR, LET e CONST passam, através de hoisting, para o topo de seu escopo.

Enquanto variáveis com VAR são inicializadas com undefined, as variáveis com LET e CONST não são inicializadas.



	VAR	LET	CONST
Escopo GLOBAL	X		
Escopo FUNÇÃO	X	X	X
Escopo BLOCO		X	X
ATUALIZAR / REDEFINIR	X	X	
REDECLARAR	X		
HOISTING	X	X	X

# VAR, LET e CONST - RESUMO

VAR tem escopo global ou de função enquanto LET e CONST têm escopo de bloco.

VAR pode ser atualizada e declarada novamente dentro de seu escopo.

LET pode ser atualizada, mas não pode ser declaradas novamente.

CONST não podem ser atualizadas nem declaradas novamente.

VAR, LET e CONST passam por *hoisting*.


Enquanto VAR é inicializada com *undefined*, LET e CONST não são inicializadas.

VAR e LET podem ser declaradas sem serem inicializadas

CONST precisa da inicialização durante a declaração.




# var

var apple = 



uma coisa numa  
caixa chamada  
"apple"

apple = 



uma coisa numa  
caixa chamada  
"apple"

# let

let apple = 



uma coisa numa  
caixa chamada  
"apple" com  
escudo protetor



you can alter the  
item only if you  
do this inside the  
shield

# const

const apple = 



uma coisa em  
uma jaula  
chamada "apple"



you can't alter  
the item after

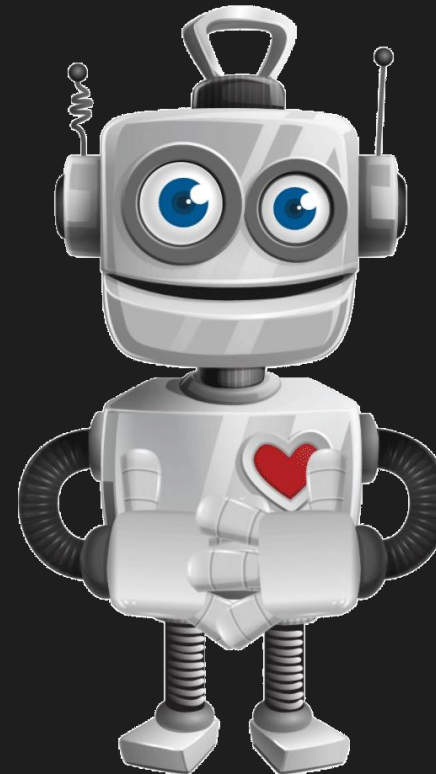


apple.multiply(3)  
ok!

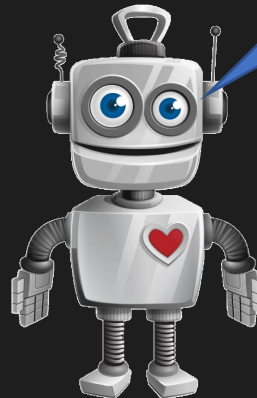
... but you can request  
that the item changes  
(if the item has a  
method to do this)

```
JS vaga.js > ...  
1  /* Elaborar um programa que leia nome, idade e pretensão salarial de um candidato a uma vaga de emprego.  
2  Caso a idade esteja entre 20 e 30 anos e a pretensão salarial entre 2000.00 e 3000.00  
3  exiba "<nome> você foi selecionado" ou senão, "<nome> procure outra vaga". */  
4  
5  const prompt = require("prompt-sync")()  
6  
7  let nome = (prompt("Nome: "))  
8  let idade = Number(prompt("Idade: "))  
9  let pret = Number(prompt("Pretensão salarial: "))
```

... continuando.  
Estávamos  
declarando  
variáveis com **LET**

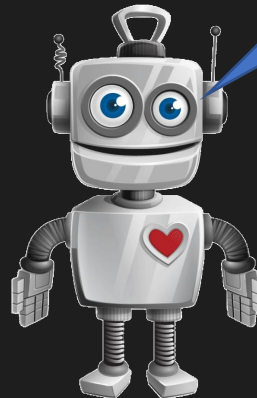


```
JS vaga.js > ...
1  /* Elaborar um programa que leia nome, idade e pretensão salarial de um candidato a uma vaga de emprego.
2  Caso a idade esteja entre 20 e 30 anos e a pretensão salarial entre 2000.00 e 3000.00
3  exiba "<nome> você foi selecionado" ou senão, "<nome> procure outra vaga". */
4
5  const prompt = require("prompt-sync")()
6
7  let nome = (prompt("Nome: "))
8  let idade = Number(prompt("Idade: "))
9  let pret = Number(prompt("Pretensão salarial: "))
10
11 if (idade >= 20 && idade <= 30 && pret >= 2000 && pret <= 3000){
```



Veja as **proposições** e **conjunções** na **condição** do IF

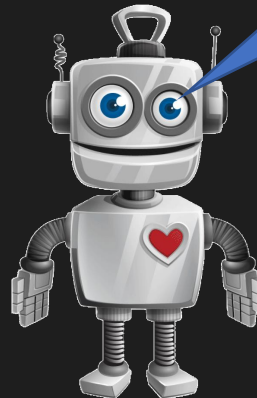
```
JS vaga.js > ...
1  /* Elaborar um programa que leia nome, idade e pretensão salarial de um candidato a uma vaga de emprego.
2  Caso a idade esteja entre 20 e 30 anos e a pretensão salarial entre 2000.00 e 3000.00
3  exiba "<nome> você foi selecionado" ou senão, "<nome> procure outra vaga". */
4
5  const prompt = require("prompt-sync")()
6
7  let nome = (prompt("Nome: "))
8  let idade = Number(prompt("Idade: "))
9  let pret = Number(prompt("Pretensão salarial: "))
10
11 if (idade >= 20 && idade <= 30 && pret >= 2000 && pret <= 3000){
```



Veja as **proposições** e **conjunções** na **condição** do IF

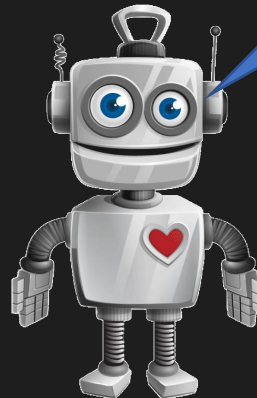
```
JS vaga.js > ...
1  /* Elaborar um programa que leia nome, idade e pretensão salarial de um candidato a uma vaga de emprego.
2  Caso a idade esteja entre 20 e 30 anos e a pretensão salarial entre 2000.00 e 3000.00
3  exiba "<nome> você foi selecionado" ou senão, "<nome> procure outra vaga". */
4
5  const prompt = require("prompt-sync")()
6
7  let nome = (prompt("Nome: "))
8  let idade = Number(prompt("Idade: "))
9  let pret = Number(prompt("Pretensão salarial: "))
10
11 if (idade >= 20 && idade <= 30 && pret >= 2000 && pret <= 3000){
12     // Você pode escrever assim
13     console.log("%s, você foi selecionado", nome)
```

Exemplo de sintaxe com o método  
log do console  
(%s para strings e %d para dígitos)



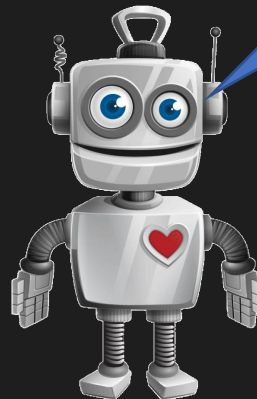
```
JS vaga.js > ...
1  /* Elaborar um programa que leia nome, idade e pretensão salarial de um candidato a uma vaga de emprego.
2  Caso a idade esteja entre 20 e 30 anos e a pretensão salarial entre 2000.00 e 3000.00
3  exiba "<nome> você foi selecionado" ou senão, "<nome> procure outra vaga". */
4
5  const prompt = require("prompt-sync")()
6
7  let nome = (prompt("Nome: "))
8  let idade = Number(prompt("Idade: "))
9  let pret = Number(prompt("Pretensão salarial: "))
10
11 if (idade >= 20 && idade <= 30 && pret >= 2000 && pret <= 3000){
12     // Você pode escrever assim
13     console.log("%s, você foi selecionado", nome)
14     // ou assado
15     console.log(`${nome}, você foi selecionado`)
```

Outro exemplo de sintaxe  
com o método log do console  
(usarei esse a partir de agora)



```
JS vaga.js > ...
1  /* Elaborar um programa que leia nome, idade e pretensão salarial de um candidato a uma vaga de emprego.
2  Caso a idade esteja entre 20 e 30 anos e a pretensão salarial entre 2000.00 e 3000.00
3  exiba "<nome> você foi selecionado" ou senão, "<nome> procure outra vaga". */
4
5  const prompt = require("prompt-sync")()
6
7  let nome = (prompt("Nome: "))
8  let idade = Number(prompt("Idade: "))
9  let pret = Number(prompt("Pretensão salarial: "))
10
11  if (idade >= 20 && idade <= 30 && pret >= 2000 && pret <= 3000){
12      // Você pode escrever assim
13      console.log("%s, você foi selecionado", nome)
14      // ou assado
15      console.log(`${nome}, você foi selecionado`)
16  } else{
17      console.log(`${nome}, procure outra vaga`)
18  }
```

Senão (executado caso a condição do if seja falsa)





```

JS vaga.js > ...
1  /* Elaborar um programa que leia nome, idade e pretensão salarial de um candidato a uma vaga de emprego.
2  Caso a idade esteja entre 20 e 30 anos e a pretensão salarial entre 2000.00 e 3000.00
3  exiba "<nome> você foi selecionado" ou senão, "<nome> procure outra vaga". */
4
5  const prompt = require("prompt-sync")()
6
7  let nome = (prompt("Nome: "))
8  let idade = Number(prompt("Idade: "))
9  let pret = Number(prompt("Pretensão salarial: "))
10
11 if (idade >= 20 && idade <= 30 && pret >= 2000 && pret <= 3000){
12     // Você pode escrever assim
13     console.log("%s, você foi selecionado", nome)
14     // ou assado
15     console.log(`${nome}, você foi selecionado`)
16 } else{
17     console.log(`${nome}, procure outra vaga`)
18 }

```

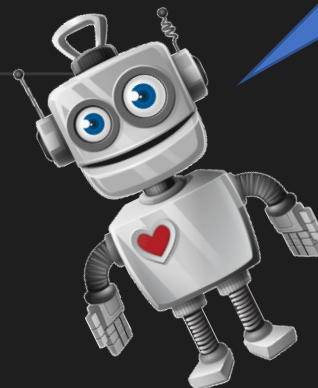
Primeiro teste

PROBLEMAS SAÍDA TERMINAL SQL CONSOLE CONSOLE DE DEPURAÇÃO

```

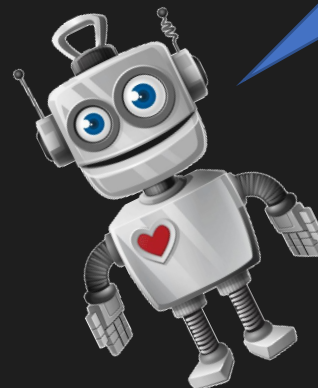
PS C:\fc> node vaga
Nome: Angelo
Idade: 26
Pretensão salarial: 2900
Angelo, você foi selecionado

```



```
JS vaga.js > ...
1  /* Elaborar um programa que leia nome, idade e pretensão salarial de um candidato a uma vaga de emprego.
2  Caso a idade esteja entre 20 e 30 anos e a pretensão salarial entre 2000.00 e 3000.00
3  exiba "<nome> você foi selecionado" ou senão, "<nome> procure outra vaga". */
4
5  const prompt = require("prompt-sync")()
6
7  let nome = (prompt("Nome: "))
8  let idade = Number(prompt("Idade: "))
9  let pret = Number(prompt("Pretensão salarial: "))
10
11  if (idade >= 20 && idade <= 30 && pret >= 2000 && pret <= 3000){
12      // Você pode escrever assim
13      console.log("%s, você foi selecionado", nome)
14      // ou assado
15      console.log(`${nome}, você foi selecionado`)
16  } else{
17      console.log(`${nome}, procure outra vaga`)
18  }
```

```
PS C:\fc> node vaga
Nome: Angelo
Idade: 26
Pretensão salarial: 3300
Angelo, procure outra vaga
```



Segundo teste  
(é o suficiente para mim)

# Exemplo1

## Desafio:

“Comente” as linhas **if** e **else** e defina um novo **if** e **else**.

Agora, o novo **if** deve conter a condição para exibir a mensagem “<nome> procure outra vaga” e o **else** “<nome> você foi selecionado”.

```
JS vaga.js > ...
1  /* Elaborar um programa que leia nome, idade e pretensão salarial de um candidato a uma vaga de emprego.
2  Caso a idade esteja entre 20 e 30 anos e a pretensão salarial entre 2000.00 e 3000.00
3  exiba "<nome> você foi selecionado" ou senão, "<nome> procure outra vaga". */
4
5  const prompt = require("prompt-sync")()
6
7  let nome = (prompt("Nome: "))
8  let idade = Number(prompt("Idade: "))
9  let pret = Number(prompt("Pretensão salarial: "))
10
11  if (idade >= 20 && idade <= 30 && pret >= 2000 && pret <= 3000){
12      console.log(`${nome}, você foi selecionado`)
13  } else{
14      console.log(`${nome}, procure outra vaga`)
15  }
```

“Comente” as linhas if e else e defina um novo if e else. Agora, o novo if deve conter a condição para exibir a mensagem “<nome> procure outra vaga” e o else “<nome> você foi selecionado”.



```
JS vaga.js > ...
1  /* Elaborar um programa que leia nome, idade e pretensão salarial de um candidato a uma vaga de emprego.
2  Caso a idade esteja entre 20 e 30 anos e a pretensão salarial entre 2000.00 e 3000.00
3  exiba "<nome> você foi selecionado" ou senão, "<nome> procure outra vaga". */
4
5  const prompt = require("prompt-sync")()
6
7  let nome = (prompt("Nome: "))
8  let idade = Number(prompt("Idade: "))
9  let pret = Number(prompt("Pretensão salarial: "))
10
11  if (idade >= 20 && idade <= 30 && pret >= 2000 && pret <= 3000){
12      console.log(`${nome}, você foi selecionado`)
13  } else{
14      console.log(`${nome}, procure outra vaga`)
15  }
```

O desafio pede para inverter estas linhas

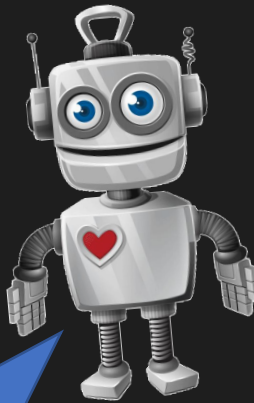


JS vaga.js > ...

```
1  /* Elaborar um programa que leia nome, idade e pretensão salarial de um candidato a uma vaga de emprego.
2  Caso a idade esteja entre 20 e 30 anos e a pretensão salarial entre 2000.00 e 3000.00
3  exiba "<nome> você foi selecionado" ou senão, "<nome> procure outra vaga". */
4
5  const prompt = require("prompt-sync")()
6
7  let nome = (prompt("Nome: "))
8  let idade = Number(prompt("Idade: "))
9  let pret = Number(prompt("Pretensão salarial: "))
10
11  if (idade >= 20 && idade <= 30 && pret >= 2000 && pret <= 3000){
12      console.log(`${nome}, você foi selecionado`)
13  } else{
14      console.log(`${nome}, procure outra vaga`)
15  }
```

Precisamos realizar a negação do &&

Precisamos fazer a inversão solicitada no desafio



Após inverter as linhas, obviamente, precisaremos negar a nossa condição

# Negação Composta

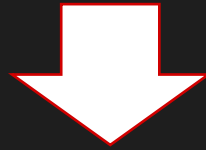
Proposição	Negação
$p \wedge q$	$\neg p \vee \neg q$
$p \vee q$	$\neg p \wedge \neg q$
$p \rightarrow q$	$p \wedge \neg q$





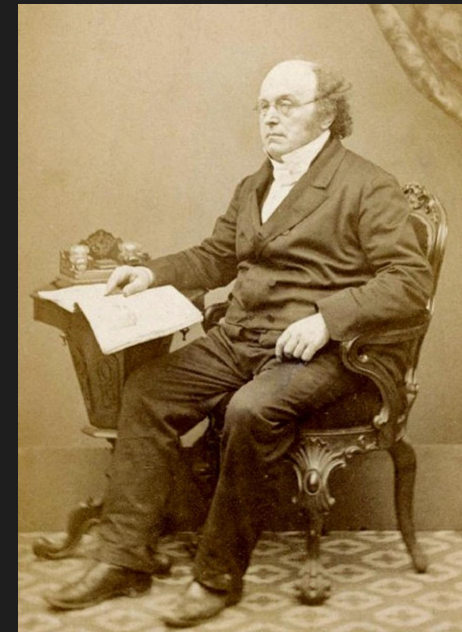
# Negação do && (^)

```
if (idade >= 20 && idade <= 30 && pret >= 2000 && pret <= 3000)
```



```
if (idade < 20 || idade > 30 || pret < 2000 || pret > 3000)
```

Proposição	Negação
$p \wedge q$	$\neg p \vee \neg q$



# Simples dicas de atalho para o VSCODE

**[ALT]+[Shift]+[↑]** – *Duplica bloco de texto selecionado (acima do local marcado)*

**[ALT]+[Shift]+[↓]** – *Duplica bloco de texto selecionado (abaixo do local marcado)*

**[CTRL]+[/]** – *“Comenta” bloco de texto selecionado (com / no início de cada linha)*

**[ALT]+[SHIFT]+[A]** – *“Comenta” bloco de texto selecionado (com /\* no início e \*/ no final)*

**[CTRL]+[S]** – *“Salva” o arquivo atual*



```
JS vaga.js > ...
1  /* Elaborar um programa que leia nome, idade e pretensão salarial de um candidato a uma vaga de emprego.
2  Caso a idade esteja entre 20 e 30 anos e a pretensão salarial entre 2000.00 e 3000.00
3  exiba "<nome> você foi selecionado" ou senão, "<nome> procure outra vaga". */
4
5  const prompt = require("prompt-sync")()
6
7  let nome = (prompt("Nome: "))
8  let idade = Number(prompt("Idade: "))
9  let pret = Number(prompt("Pretensão salarial: "))
10
11  if (idade >= 20 && idade <= 30 && pret >= 2000 && pret <= 3000){
12      console.log(`${nome}, você foi selecionado`)
13  } else{
14      console.log(`${nome}, procure outra vaga`)
15  }
```

Precisamos realizar a negação do &&

Precisamos fazer a inversão solicitada no desafio



Bora fazer?

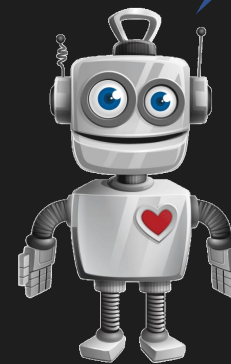
JS vaga.js > ...

```
1  /* Elaborar um programa que leia nome, idade e pretensão salarial de um candidato a uma vaga de emprego.
2  Caso a idade esteja entre 20 e 30 anos e a pretensão salarial entre 2000.00 e 3000.00
3  exiba "<nome> você foi selecionado" ou senão, "<nome> procure outra vaga". */
4
5  const prompt = require("prompt-sync")()
6
7  let nome = (prompt("Nome: "))
8  let idade = Number(prompt("Idade: "))
9  let pret = Number(prompt("Pretensão salarial: "))
10
11 // if (idade >= 20 && idade <= 30 && pret >= 2000 && pret <= 3000){
12 //     console.log(`${nome}, você foi selecionado`)
13 // } else{
14 //     console.log(`${nome}, procure outra vaga`)
15 // }
16 if (idade < 20 || idade > 30 || pret < 2000 || pret > 3000){
17     console.log(`${nome}, procure outra vaga`)
18 } else{
19     console.log(`${nome}, você foi selecionado`)
20 }
```

Negação do && realizada

Inversão feita

Tio De Morgan  
está de olho



Proposição	Negação
$p \wedge q$	$\neg p \vee \neg q$

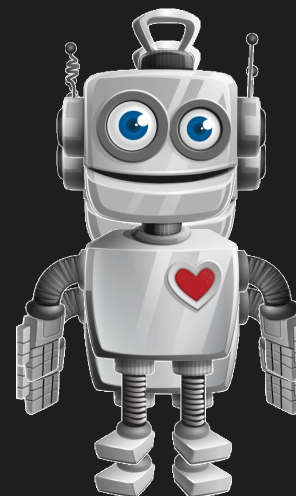


JS vaga.js > ...

```
1  /* Elaborar um programa que leia nome, idade e pretensão salarial de um candidato a uma vaga de emprego.
2  Caso a idade esteja entre 20 e 30 anos e a pretensão salarial entre 2000.00 e 3000.00
3  exiba "<nome> você foi selecionado" ou senão, "<nome> procure outra vaga". */
4
5  const prompt = require("prompt-sync")()
6
7  let nome = (prompt("Nome: "))
8  let idade = Number(prompt("Idade: "))
9  let pret = Number(prompt("Pretensão salarial: "))
10
11 // if (idade >= 20 && idade <= 30 && pret >= 2000 && pret <= 3000){
12 //     console.log(`${nome}, você foi selecionado`)
13 // } else{
14 //     console.log(`${nome}, procure outra vaga`)
15 // }
16 if (idade < 20 || idade > 30 || pret < 2000 || pret > 3000){
17     console.log(`${nome}, procure outra vaga`)
18 } else{
19     console.log(`${nome}, você foi selecionado`)
20 }
```

Em 1870 eu já estava careca  
de saber que isso iria  
funcionar.  
E nem usava o VSCode

Chupa, ChatGPT!



# Exemplo2

Elaborar um programa que leia **3 lados** e verifique se eles podem ou não formar um **triângulo**.

Para formar um triângulo um dos lados não pode ser maior que a soma dos outros dois.

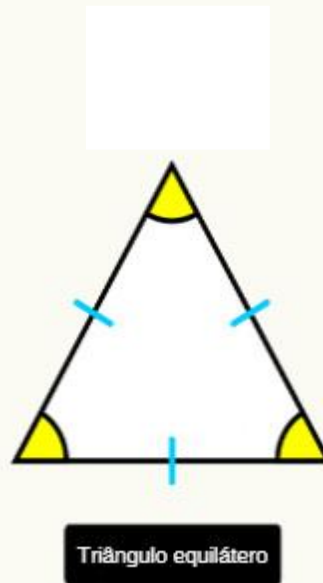
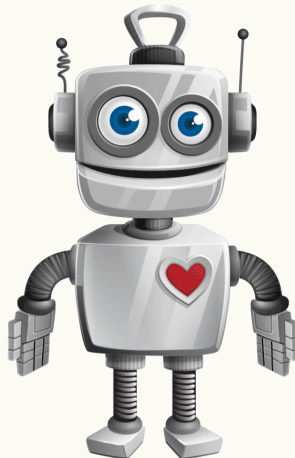
# Exemplo2

Elaborar um programa que leia **3 lados** e verifique se eles podem ou não formar um **triângulo**.

Para formar um triângulo um dos lados não pode ser maior que a soma dos outros dois.

Caso possam formar, apresente também qual o tipo de triângulo:

- **Equilátero:** 3 lados iguais





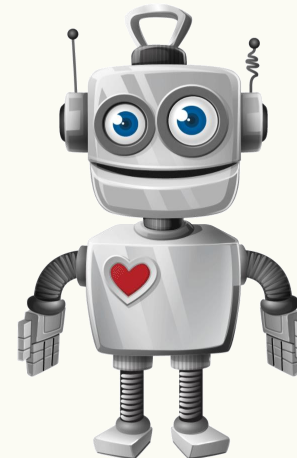
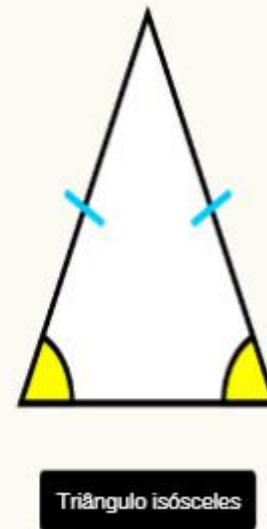
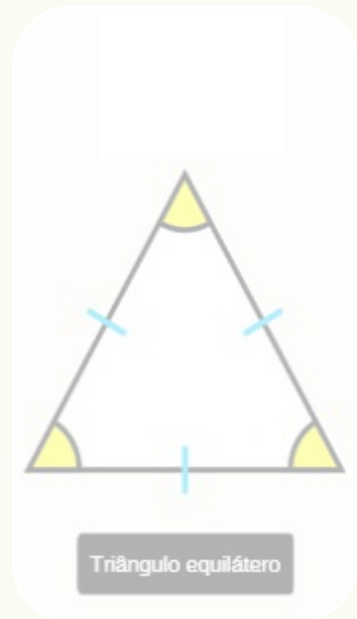
# Exemplo2

Elaborar um programa que leia **3 lados** e verifique se eles podem ou não formar um **triângulo**.

Para formar um triângulo um dos lados não pode ser maior que a soma dos outros dois.

Caso possam formar, apresente também qual o tipo de triângulo:

- **Equilátero:** 3 lados iguais
- **Isósceles:** 2 lados iguais



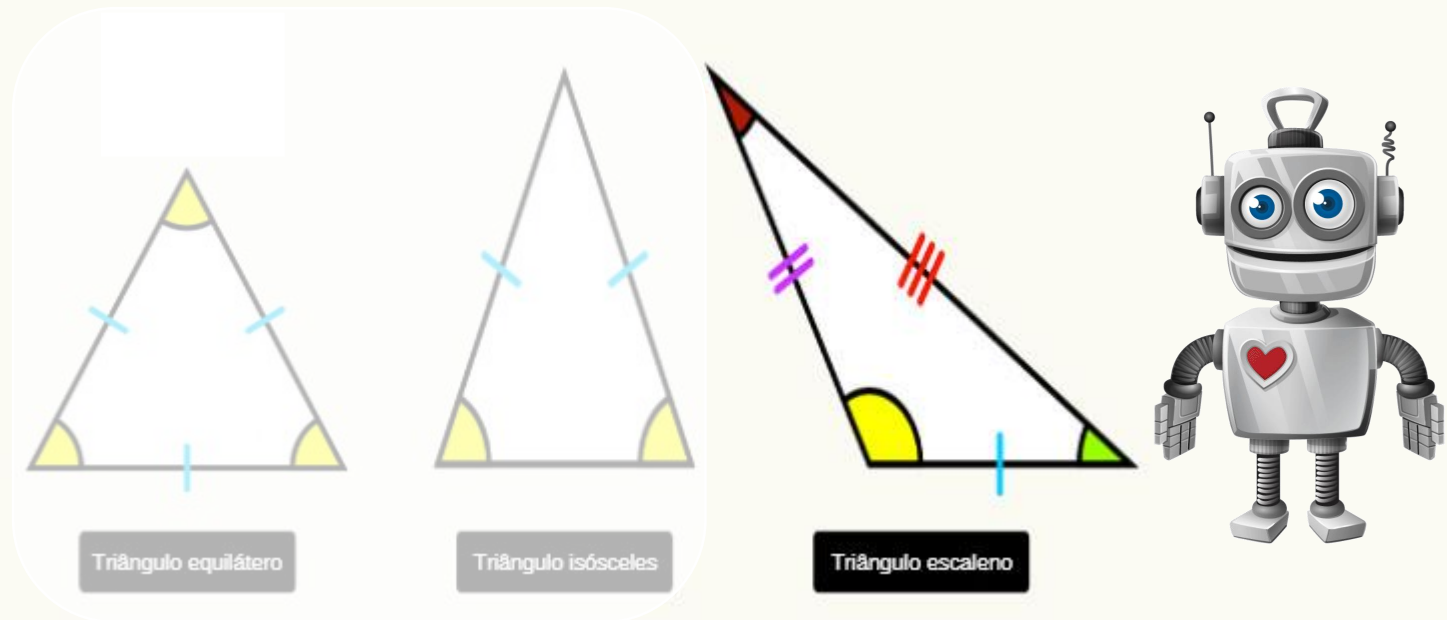
# Exemplo2

Elaborar um programa que leia **3 lados** e verifique se eles podem ou não formar um **triângulo**.

Para formar um triângulo um dos lados não pode ser maior que a soma dos outros dois.

Caso possam formar, apresente também qual o tipo de triângulo:

- **Equilátero:** 3 lados iguais
- **Isósceles:** 2 lados iguais
- **Escaleno:** 3 lados diferentes



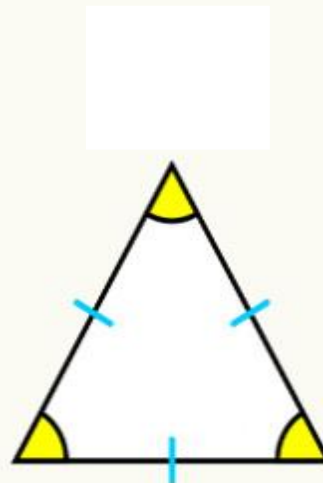
# Exemplo2

Elaborar um programa que leia **3 lados** e verifique se eles podem ou não formar um **triângulo**.

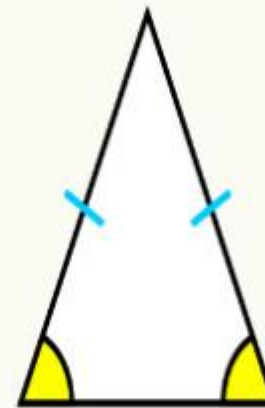
Para formar um triângulo um dos lados não pode ser maior que a soma dos outros dois.

Caso possam formar, apresente também qual o tipo de triângulo:

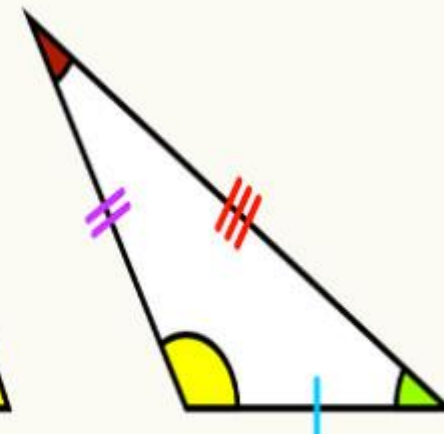
- **Equilátero:** 3 lados iguais
- **Isósceles:** 2 lados iguais
- **Escaleno:** 3 lados diferentes



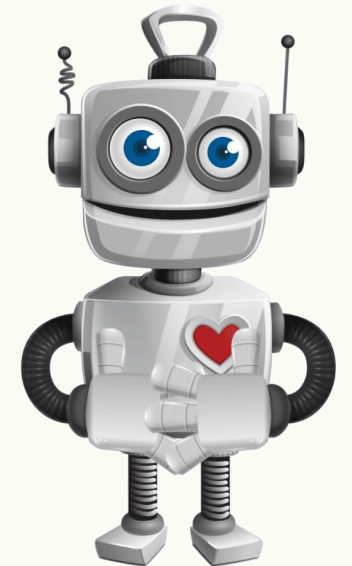
Triângulo equilátero



Triângulo isósceles



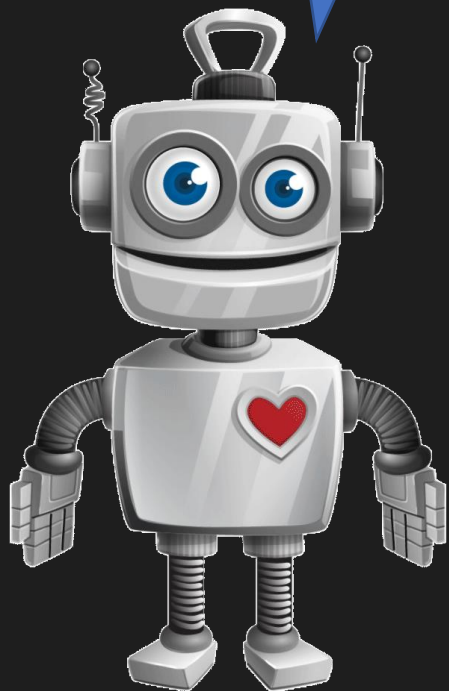
Triângulo escaleno





**TRIÂNGULO DAS BERMUDAS**

Tio Glad achou a piada engraçada  
e mandou rir.  
Eu estou rindo muito (por dentro)

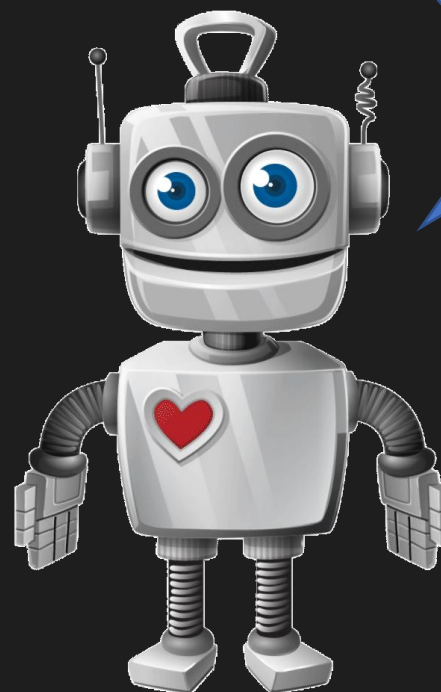




JS triangulo.js &gt; [🔍] prompt

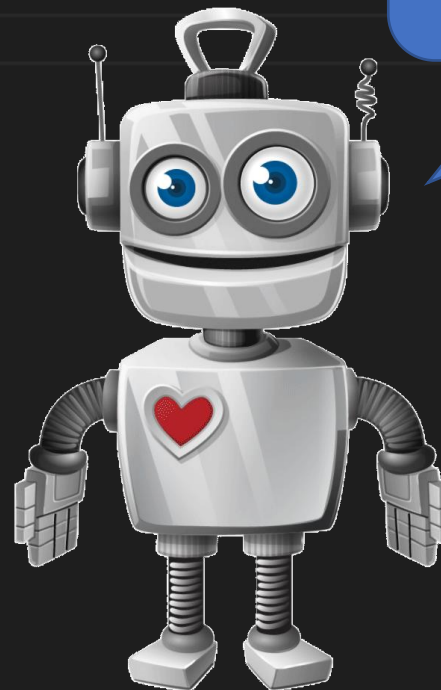
```
1 // Elaborar um programa que leia 3 lados e verifique se eles podem ou não formar um triângulo.  
2 // Para formar um triângulo um dos lados não pode ser maior que a soma dos outros dois.  
3  
4 // Caso possam formar, apresente também qual o tipo de triângulo:  
5 // - Equilátero: 3 lados iguais  
6 // - Isósceles: 2 lados iguais  
7 // - Escaleno: 3 lados diferentes
```

É tão bom colocar o  
enunciado junto ao Código  
(poupa vários ALT+TAB)



JS triangulo.js &gt; [🔗] prompt

```
1 // Elaborar um programa que leia 3 lados e verifique se eles podem ou não formar um triângulo.  
2 // Para formar um triângulo um dos lados não pode ser maior que a soma dos outros dois.  
3  
4 // Caso possam formar, apresente também qual o tipo de triângulo:  
5 // - Equilátero: 3 lados iguais  
6 // - Isósceles: 2 lados iguais  
7 // - Escaleno: 3 lados diferentes  
8  
9 const prompt = require("prompt-sync")()
```



prompt-sync, meu precioso!



JS triangulo.js &gt; [?] prompt

```
1 // Elaborar um programa que leia 3 lados e verifique se eles podem ou não formar um triângulo.
2 // Para formar um triângulo um dos lados não pode ser maior que a soma dos outros dois.
3
4 // Caso possam formar, apresente também qual o tipo de triângulo:
5 // - Equilátero: 3 lados iguais
6 // - Isósceles: 2 lados iguais
7 // - Escaleno: 3 lados diferentes
8
9 const prompt = require("prompt-sync")()
10
11 let a = Number(prompt("Lado A: "))
12 let b = Number(prompt("Lado B: "))
13 let c = Number(prompt("Lado C: "))
```

Esses nomes de variáveis  
foram inspirados no  
minimalismo, né?  
Fala sério!



JS triangulo.js &gt; [🔗] prompt

```
1 // Elaborar um programa que leia 3 lados e verifique se eles podem ou não formar um triângulo.
2 // Para formar um triângulo um dos lados não pode ser maior que a soma dos outros dois.
3
4 // Caso possam formar, apresente também qual o tipo de triângulo:
5 // - Equilátero: 3 lados iguais
6 // - Isósceles: 2 lados iguais
7 // - Escaleno: 3 lados diferentes
8
9 const prompt = require("prompt-sync")()
10
11 let a = Number(prompt("Lado A: "))
12 let b = Number(prompt("Lado B: "))
13 let c = Number(prompt("Lado C: "))
14
15 if(a > b+c || b > a+c || c > a+b){
```

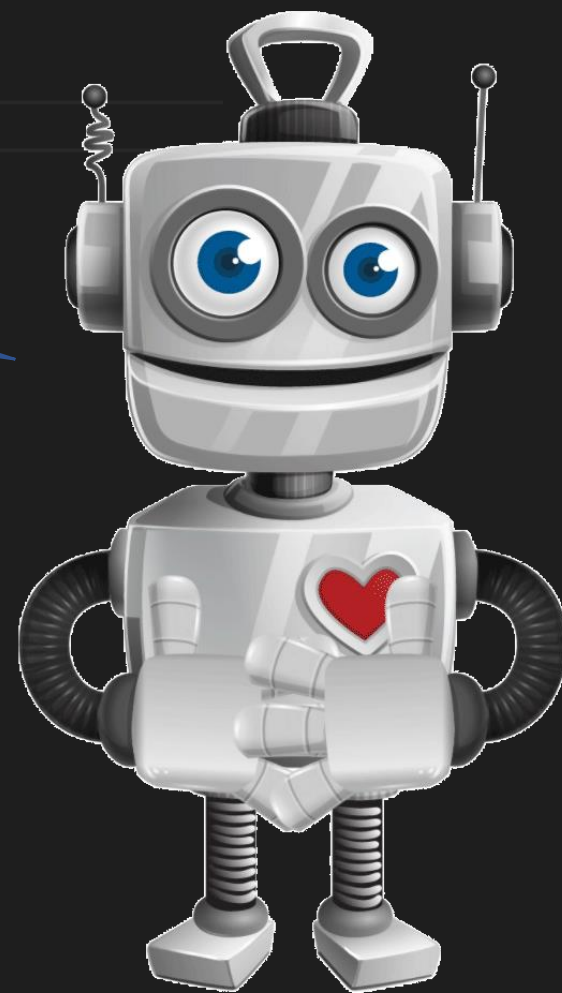
Uma condição dessas, bicho!



JS triangulo.js &gt; [🔗] prompt

```
1 // Elaborar um programa que leia 3 lados e verifique se eles podem ou não formar um triângulo.
2 // Para formar um triângulo um dos lados não pode ser maior que a soma dos outros dois.
3
4 // Caso possam formar, apresente também qual o tipo de triângulo:
5 // - Equilátero: 3 lados iguais
6 // - Isósceles: 2 lados iguais
7 // - Escaleno: 3 lados diferentes
8
9 const prompt = require("prompt-sync")()
10
11 let a = Number(prompt("Lado A: "))
12 let b = Number(prompt("Lado B: "))
13 let c = Number(prompt("Lado C: "))
14
15 if(a > b+c || b > a+c || c > a+b){
16     console.log("Lados NÃO formam um triângulo")
17 }
```

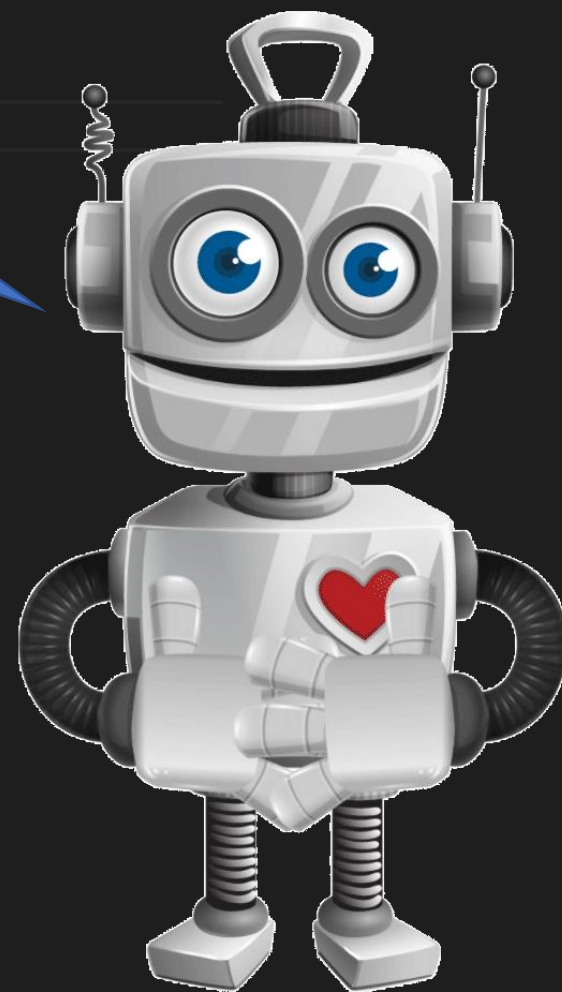
Eu já disse que se um dos lados for maior do que a soma dos outros dois então não formará um triângulo?



JS triangulo.js &gt; [🔍] prompt

```
1 // Elaborar um programa que leia 3 lados e verifique se eles podem ou não formar um triângulo.
2 // Para formar um triângulo um dos lados não pode ser maior que a soma dos outros dois.
3
4 // Caso possam formar, apresente também qual o tipo de triângulo:
5 // - Equilátero: 3 lados iguais
6 // - Isósceles: 2 lados iguais
7 // - Escaleno: 3 lados diferentes
8
9 const prompt = require("prompt-sync")()
10
11 let a = Number(prompt("Lado A: "))
12 let b = Number(prompt("Lado B: "))
13 let c = Number(prompt("Lado C: "))
14
15 if(a > b+c || b > a+c || c > a+b){
16   console.log("Lados NÃO formam um triângulo")
17 }else{
18   console.log("Lados podem formar um triângulo")
```

Se a condição for falsa então  
é pq forma triângulo.



JS triangulo.js &gt; [🔍] prompt

```
1 // Elaborar um programa que leia 3 lados e verifique se eles podem ou não formar um triângulo.
2 // Para formar um triângulo um dos lados não pode ser maior que a soma dos outros dois.
3
4 // Caso possam formar, apresente também qual o tipo de triângulo:
5 // - Equilátero: 3 lados iguais
6 // - Isósceles: 2 lados iguais
7 // - Escaleno: 3 lados diferentes
8
9 const prompt = require("prompt-sync")()
10
11 let a = Number(prompt("Lado A: "))
12 let b = Number(prompt("Lado B: "))
13 let c = Number(prompt("Lado C: "))
14
15 if(a > b+c || b > a+c || c > a+b){
16     console.log("Lados NÃO formam um triângulo")
17 }else{
18     console.log("Lados podem formar um triângulo")
19     if (a==b && b==c){
```

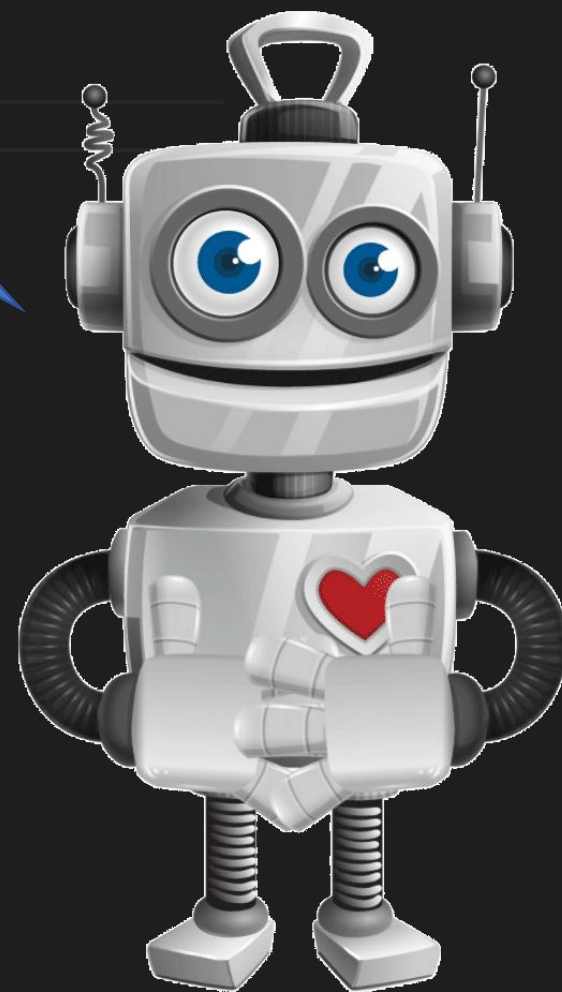
Três lados iguais



JS triangulo.js &gt; [🔍] prompt

```
1 // Elaborar um programa que leia 3 lados e verifique se eles podem ou não formar um triângulo.
2 // Para formar um triângulo um dos lados não pode ser maior que a soma dos outros dois.
3
4 // Caso possam formar, apresente também qual o tipo de triângulo:
5 // - Equilátero: 3 lados iguais
6 // - Isósceles: 2 lados iguais
7 // - Escaleno: 3 lados diferentes
8
9 const prompt = require("prompt-sync")()
10
11 let a = Number(prompt("Lado A: "))
12 let b = Number(prompt("Lado B: "))
13 let c = Number(prompt("Lado C: "))
14
15 if(a > b+c || b > a+c || c > a+b){
16     console.log("Lados NÃO formam um triângulo")
17 }else{
18     console.log("Lados podem formar um triângulo")
19     if (a==b && b==c){
20         console.log("Triângulo equilátero")
```

Triângulo  
equilátero

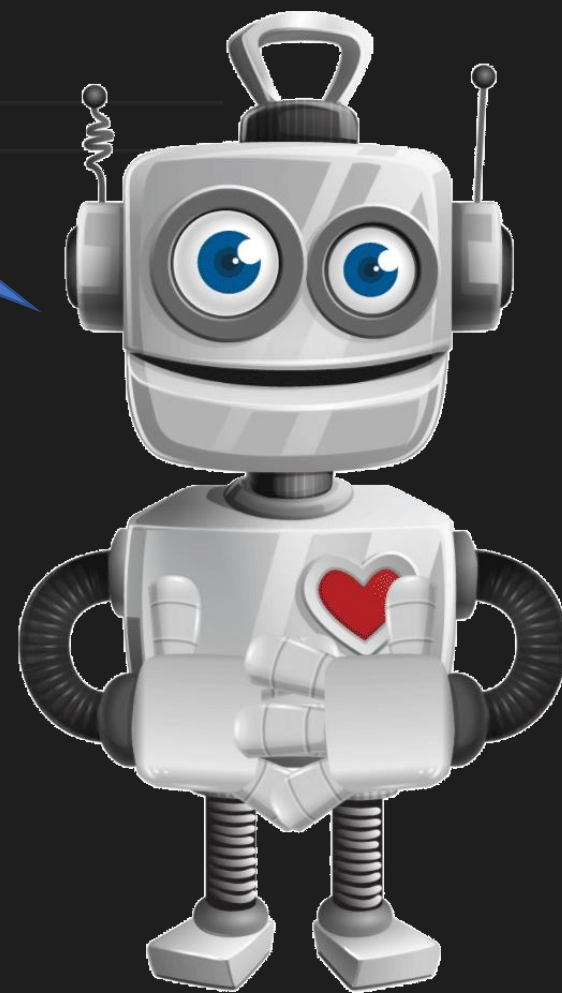




JS triangulo.js &gt; [x] prompt

```
1 // Elaborar um programa que leia 3 lados e verifique se eles podem ou não formar um triângulo.
2 // Para formar um triângulo um dos lados não pode ser maior que a soma dos outros dois.
3
4 // Caso possam formar, apresente também qual o tipo de triângulo:
5 // - Equilátero: 3 lados iguais
6 // - Isósceles: 2 lados iguais
7 // - Escaleno: 3 lados diferentes
8
9 const prompt = require("prompt-sync")()
10
11 let a = Number(prompt("Lado A: "))
12 let b = Number(prompt("Lado B: "))
13 let c = Number(prompt("Lado C: "))
14
15 if(a > b+c || b > a+c || c > a+b){
16     console.log("Lados NÃO formam um triângulo")
17 }else{
18     console.log("Lados podem formar um triângulo")
19     if (a==b && b==c){
20         console.log("Triângulo equilátero")
21     } else if (a==b || a==c || b==c) {
```

Não possuí 3 lados iguais  
Mas possuí 2 lados iguais





JS triangulo.js &gt; [🔍] prompt

```
1 // Elaborar um programa que leia 3 lados e verifique se eles podem ou não formar um triângulo.
2 // Para formar um triângulo um dos lados não pode ser maior que a soma dos outros dois.
3
4 // Caso possam formar, apresente também qual o tipo de triângulo:
5 // - Equilátero: 3 lados iguais
6 // - Isósceles: 2 lados iguais
7 // - Escaleno: 3 lados diferentes
8
9 const prompt = require("prompt-sync")()
10
11 let a = Number(prompt("Lado A: "))
12 let b = Number(prompt("Lado B: "))
13 let c = Number(prompt("Lado C: "))
14
15 if(a > b+c || b > a+c || c > a+b){
16     console.log("Lados NÃO formam um triângulo")
17 }else{
18     console.log("Lados podem formar um triângulo")
19     if (a==b && b==c){
20         console.log("Triângulo equilátero")
21     } else if (a==b || a==c || b==c) {
22         console.log("Triângulo isósceles")
23     }
24 }
```

Triângulo  
isósceles



JS triangulo.js &gt; [?] prompt

```
1 // Elaborar um programa que leia 3 lados e verifique se eles podem ou não formar um triângulo.
2 // Para formar um triângulo um dos lados não pode ser maior que a soma dos outros dois.
3
4 // Caso possam formar, apresente também qual o tipo de triângulo:
5 // - Equilátero: 3 lados iguais
6 // - Isósceles: 2 lados iguais
7 // - Escaleno: 3 lados diferentes
8
9 const prompt = require("prompt-sync")()
10
11 let a = Number(prompt("Lado A: "))
12 let b = Number(prompt("Lado B: "))
13 let c = Number(prompt("Lado C: "))
14
15 if(a > b+c || b > a+c || c > a+b){
16     console.log("Lados NÃO formam um triângulo")
17 }else{
18     console.log("Lados podem formar um triângulo")
19     if (a==b && b==c){
20         console.log("Triângulo equilátero")
21     } else if (a==b || a==c || b==c) {
22         console.log("Triângulo isósceles")
23     } else{
```

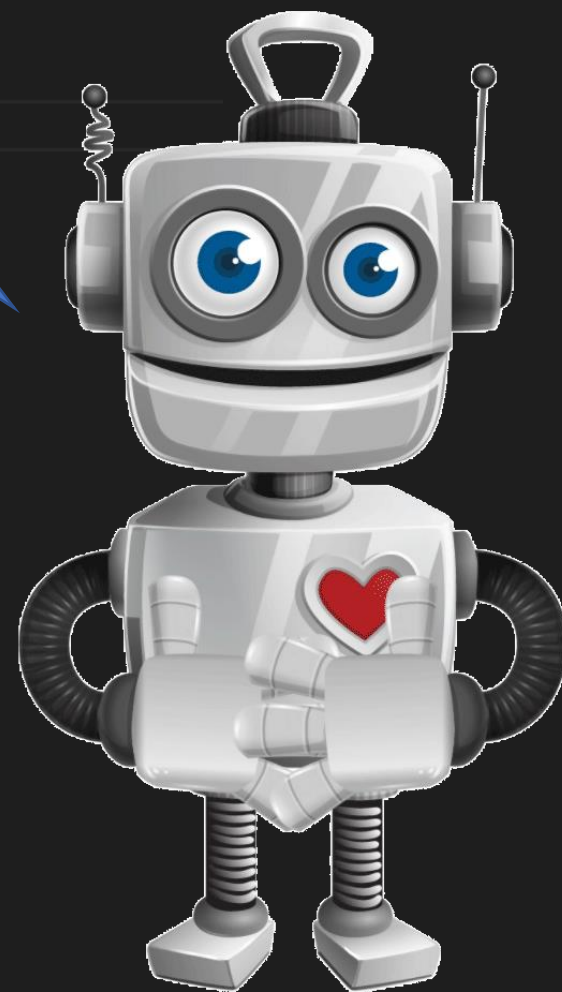
Caso contrário?



JS triangulo.js &gt; [?] prompt

```
1 // Elaborar um programa que leia 3 lados e verifique se eles podem ou não formar um triângulo.
2 // Para formar um triângulo um dos lados não pode ser maior que a soma dos outros dois.
3
4 // Caso possam formar, apresente também qual o tipo de triângulo:
5 // - Equilátero: 3 lados iguais
6 // - Isósceles: 2 lados iguais
7 // - Escaleno: 3 lados diferentes
8
9 const prompt = require("prompt-sync")()
10
11 let a = Number(prompt("Lado A: "))
12 let b = Number(prompt("Lado B: "))
13 let c = Number(prompt("Lado C: "))
14
15 if(a > b+c || b > a+c || c > a+b){
16     console.log("Lados NÃO formam um triângulo")
17 }else{
18     console.log("Lados podem formar um triângulo")
19     if (a==b && b==c){
20         console.log("Triângulo equilátero")
21     } else if (a==b || a==c || b==c) {
22         console.log("Triângulo isósceles")
23     } else{
24         console.log("Triângulo escaleno")
25     }
26 }
```

Triângulo  
Escaleno



JS triangulo.js &gt; [?] prompt

```
1 // Elaborar um programa que leia 3 lados e verifique se eles podem ou não formar um triângulo.
2 // Para formar um triângulo um dos lados não pode ser maior que a soma dos outros dois.
3
4 // Caso possam formar, apresente também qual o tipo de triângulo:
5 // - Equilátero: 3 lados iguais
6 // - Isósceles: 2 lados iguais
7 // - Escaleno: 3 lados diferentes
8
9 const prompt = require("prompt-sync")()
10
11 let a = Number(prompt("Lado A: "))
12 let b = Number(prompt("Lado B: "))
13 let c = Number(prompt("Lado C: "))
14
15 if(a > b+c || b > a+c || c > a+b){
16     console.log("Lados NÃO formam um triângulo")
17 }else{
18     console.log("Lados podem formar um triângulo")
19     if (a==b && b==c){
20         console.log("Triângulo equilátero")
21     } else if (a==b || a==c || b==c) {
22         console.log("Triângulo isósceles")
23     } else{
24         console.log("Triângulo escaleno")
25     }
26 }
```

Teste 1

PROBLEMAS

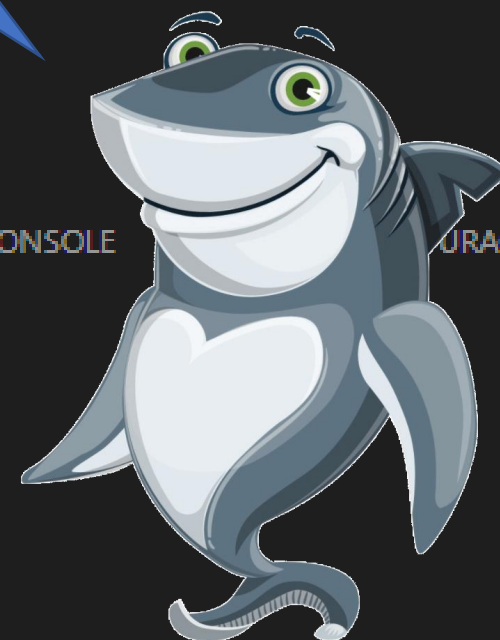
SAÍDA

TERMINAL

SQL CONSOLE

CRIAÇÃO

```
PS C:\fc> node triangulo
Lado A: 9
Lado B: 9
Lado C: 9
Lados podem formar um triângulo
Triângulo: Equilátero
PS C:\fc> 
```



JS triangulo.js &gt; [🔍] prompt

```
1 // Elaborar um programa que leia 3 lados e verifique se eles podem ou não formar um triângulo.
2 // Para formar um triângulo um dos lados não pode ser maior que a soma dos outros dois.
3
4 // Caso possam formar, apresente também qual o tipo de triângulo:
5 // - Equilátero: 3 lados iguais
6 // - Isósceles: 2 lados iguais
7 // - Escaleno: 3 lados diferentes
8
9 const prompt = require("prompt-sync")()
10
11 let a = Number(prompt("Lado A: "))
12 let b = Number(prompt("Lado B: "))
13 let c = Number(prompt("Lado C: "))
14
15 if(a > b+c || b > a+c || c > a+b){
16     console.log("Lados NÃO formam um triângulo")
17 }else{
18     console.log("Lados podem formar um triângulo")
19     if (a==b && b==c){
20         console.log("Triângulo equilátero")
21     } else if (a==b || a==c || b==c) {
22         console.log("Triângulo isósceles")
23     } else{
24         console.log("Triângulo escaleno")
25     }
26 }
```

Teste 2

PROBLEMAS

SAÍDA

TERMINAL

SQL CONSOLE

DEPURAÇÃO

PS C:\fc&gt; node triangulo

Lado A: 9

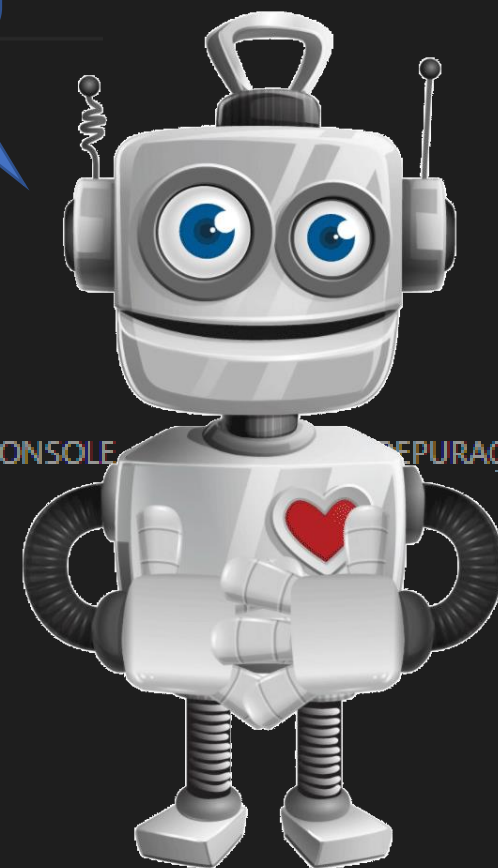
Lado B: 6

Lado C: 6

Lados podem formar um triângulo

Triângulo: Isósceles

PS C:\fc&gt; █





JS triangulo.js &gt; [?] prompt

```
1 // Elaborar um programa que leia 3 lados e verifique se eles podem ou não formar um triângulo.
2 // Para formar um triângulo um dos lados não pode ser maior que a soma dos outros dois.
3
4 // Caso possam formar, apresente também qual o tipo de triângulo:
5 // - Equilátero: 3 lados iguais
6 // - Isósceles: 2 lados iguais
7 // - Escaleno: 3 lados diferentes
8
9 const prompt = require("prompt-sync")()
10
11 let a = Number(prompt("Lado A: "))
12 let b = Number(prompt("Lado B: "))
13 let c = Number(prompt("Lado C: "))
14
15 if(a > b+c || b > a+c || c > a+b){
16     console.log("Lados NÃO formam um triângulo")
17 }else{
18     console.log("Lados podem formar um triângulo")
19     if (a==b && b==c){
20         console.log("Triângulo equilátero")
21     } else if (a==b || a==c || b==c) {
22         console.log("Triângulo isósceles")
23     } else{
24         console.log("Triângulo escaleno")
25     }
26 }
```

Teste 3

PROBLEMAS

SAÍDA

TERMINAL

SQL CONSOLE

DURAÇÃO

PS C:\fc&gt; node triangulo

Lado A: 9

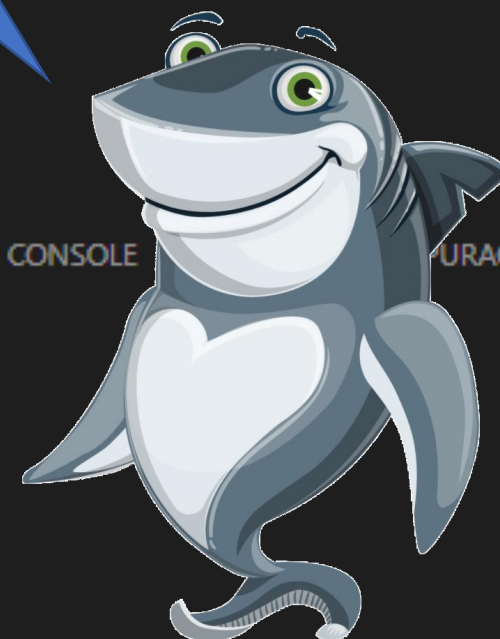
Lado B: 6

Lado C: 4

Lados podem formar um triângulo

Triângulo: Escaleno

PS C:\fc&gt; █



JS triangulo.js &gt; [?] prompt

```
1 // Elaborar um programa que leia 3 lados e verifique se eles podem ou não formar um triângulo.
2 // Para formar um triângulo um dos lados não pode ser maior que a soma dos outros dois.
3
4 // Caso possam formar, apresente também qual o tipo de triângulo:
5 // - Equilátero: 3 lados iguais
6 // - Isósceles: 2 lados iguais
7 // - Escaleno: 3 lados diferentes
8
9 const prompt = require("prompt-sync")()
10
11 let a = Number(prompt("Lado A: "))
12 let b = Number(prompt("Lado B: "))
13 let c = Number(prompt("Lado C: "))
14
15 if(a > b+c || b > a+c || c > a+b){
16     console.log("Lados NÃO formam um triângulo")
17 }else{
18     console.log("Lados podem formar um triângulo")
19     if (a==b && b==c){
20         console.log("Triângulo equilátero")
21     } else if (a==b || a==c || b==c) {
22         console.log("Triângulo isósceles")
23     } else{
24         console.log("Triângulo escaleno")
25     }
26 }
```

Teste 4

PROBLEMAS

SAÍDA

TERMINAL

SQL CONSOLE

CONSOLE DE PURAÇÃO

PS C:\fc&gt; node triangulo

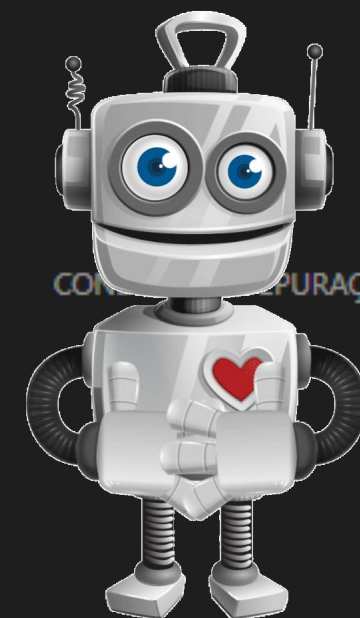
Lado A: 9

Lado B: 6

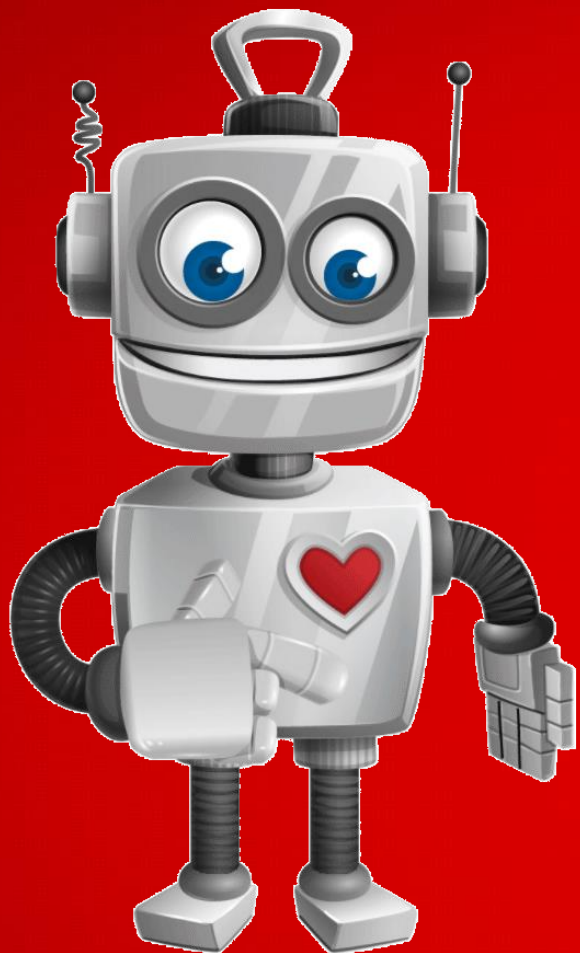
Lado C: 20

Lados NÃO podem formar um triângulo...

PS C:\fc&gt; █







**BAZINGA!**  
**Sheldonmir Cooperino**

Se curtiu, deixa o like maroto, te inscreve no canal, ativa as notificações e não compartilha com os amigos não (eles que se matriculem na Faculdade também)



TUA VEZ!

# 3

3. Uma empresa quer verificar se um empregado está qualificado para a aposentadoria ou não. Para estar em condições, um dos seguintes requisitos deve ser satisfeito:

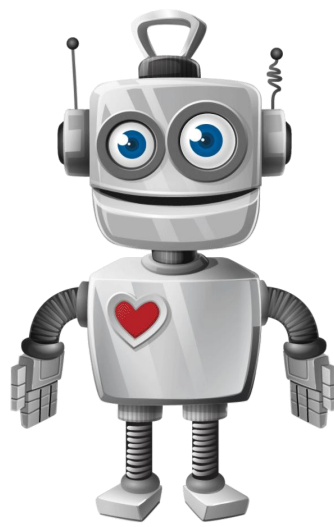
- Ter no mínimo 60 anos de idade.
- Ter trabalhado no mínimo 25 anos.
- Ter no mínimo 55 anos e ter trabalhado no mínimo 20 anos.

Com base nas informações acima, faça um algoritmo que leia nome, ano de nascimento e ano que o empregado começou a trabalhar.

Exibir a idade, tempo de trabalho e a mensagem 'pode requerer aposentadoria' ou 'não pode requerer'.

**Nome:** Gilberto  
**Ano de Nascimento:** 1962  
**Ano que começou a trabalhar:** 1982  
**Idade:** 62  
**Tempo de Serviço:** 42  
*Gilberto, você pode requerer aposentadoria*

*Obs.: Considerar ano atual = 2024*



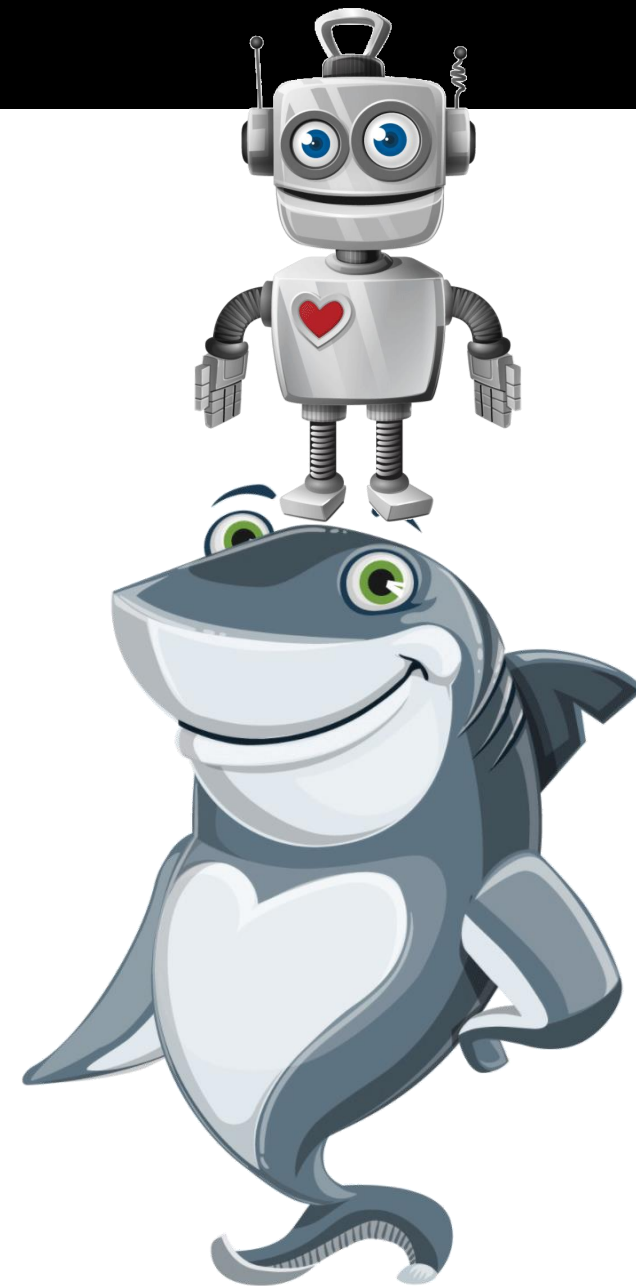
4. Elaborar um programa que leia 3 números (diferentes).  
Calcule e exiba a soma dos 2 maiores.

**1º Número:** 8

**2º Número:** 7

**3º Número:** 10

**Soma dos 2 maiores é:** 18



5. Elaborar um programa que leia 4 números (diferentes).  
Calcule e exiba a multiplicação dos 2 menores.

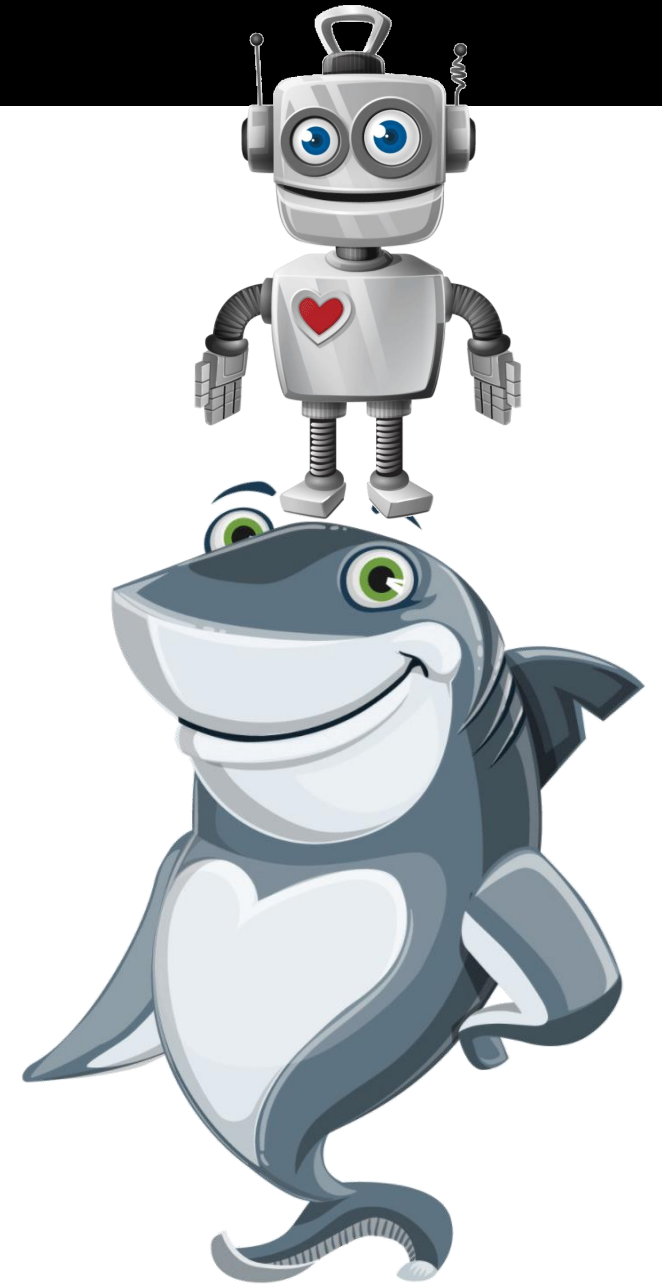
**1º Número:** 2

**2º Número:** 10

**3º Número:** 3

**3º Número:** 8

**A multiplicação dos 2 menores é:** 6

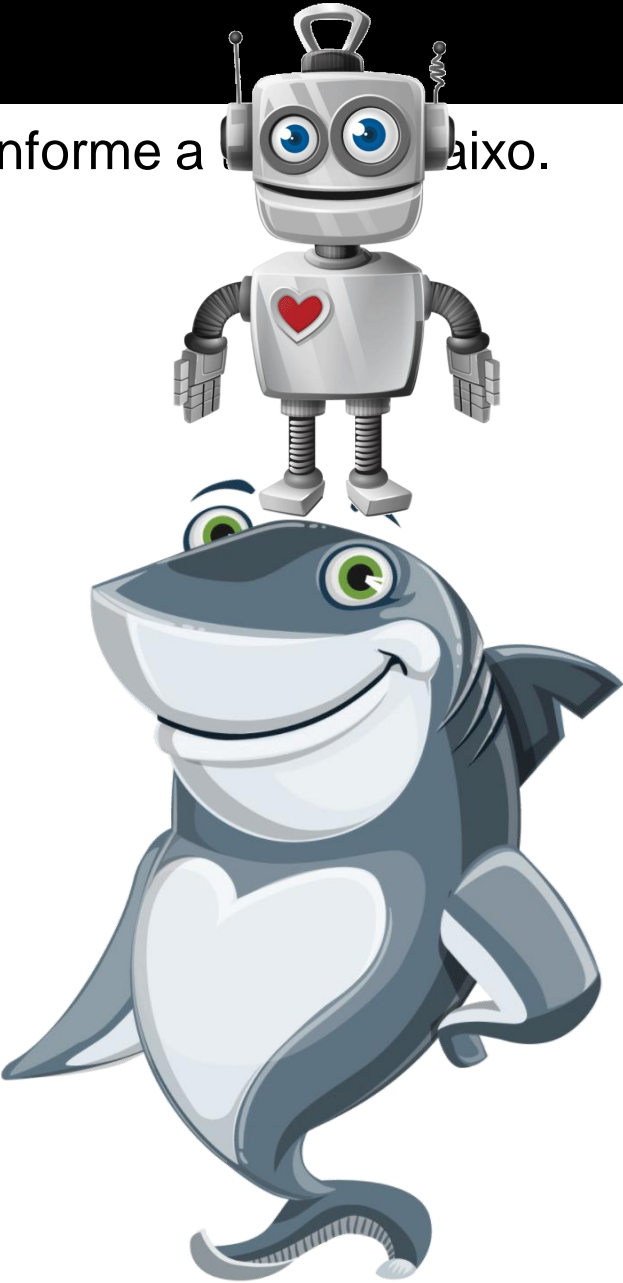




# 6

6. Faça um programa que leia a nota de um aluno e atribua um conceito conforme a tabela abaixo. Exiba a nota e o conceito correspondente.

Nota $\geq 90$	: Conceito A
Nota $\geq 80$ e $< 90$	: Conceito B
Nota $\geq 70$ e $< 80$	: Conceito C
Nota $\geq 60$ e $< 70$	: Conceito D
Nota $< 60$	: Conceito F



7. Faça um programa que leia o peso e a altura de uma pessoa, calcule o Índice de Massa Corporal (IMC) e exiba a classificação conforme a tabela abaixo.

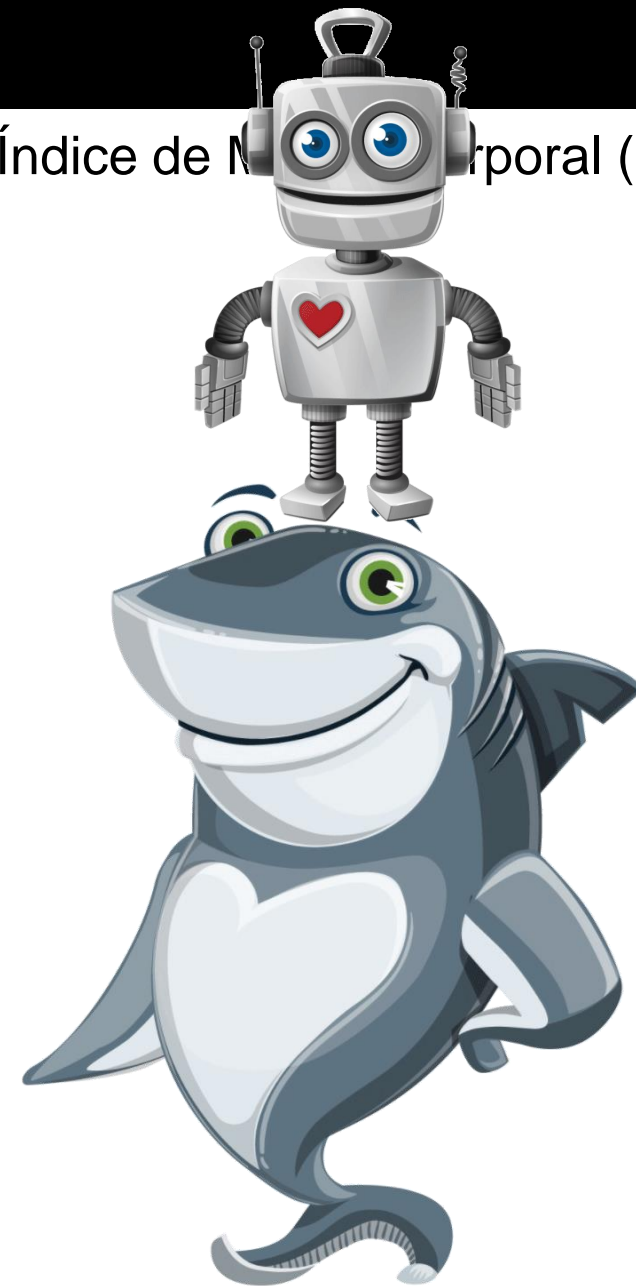
A fórmula para calcular o IMC é:  $IMC = peso / (altura * altura)$ .

**IMC < 18.5 : Abaixo do peso**

**18.5 <= IMC < 24.9 : Peso normal**

**25 <= IMC < 29.9 : Sobrepeso**

**IMC >= 30 : Obesidade**





# 8

## 8. Verificar Se Um Número é Par ou Ímpar

Faça um programa que leia um número e exiba uma mensagem indicando se ele é par ou ímpar.

